The CyberGnome

A Revolution in Learning

White Paper

4310 4306-003

© Unisys Corporation

The CyberGnome
A Revolution in Learning

The Next Revolution of Business—Mechanized Learning

Executive Summary

The CyberGnome is a learning machine that automatically creates and maintains knowledge by interacting with people and/or information systems. This white paper explains how the CyberGnome can help your business, how it learns, and how it is implemented.

The CyberGnome is a revolutionary new technology that can help break down all the barriers to your success.

Introduction

From experience, you know that you're in an aggressive, dynamically changing market. Just staying in place is difficult enough—getting ahead is becoming harder and harder. Your main concern may be trying to determine how to get a larger piece of the market or finding new markets to carve out, all while trying to keep your ever escalating costs down.

Then, if that's not enough, you're continuously challenged to ensure that your products appear obviously different from the competition, while staying current with the never-ending hot new trends that buffet the industry. And, while doing all of that, you're always looking for ways to improve your margins and decrease your costs of operation.

Increasing Market Share

One of today's major challenges, for most businesses, is increasing market share. It's not enough to be a player in a market; you need to constantly figure out ways to increase your presence and market share. One way is by extending your existing solutions beyond their current functionality. The CyberGnome can significantly extend solution functionality by learning and gaining knowledge about your solution. It can uncover new and exciting ways to enhance and amplify your solution set—without having anyone reengineer your product. You can actually take your existing solution to new levels of capability that were not possible before now, supplying experience and knowledge that previously came only from a skilled user of the solution.

Reducing your Sales Cycle

By delivering superior functionality and advanced technology that no one else in your market segment can deliver, you will help create industry demand, which in turn will help reduce your sales cycle. Plus, with the embedded CyberGnome technology, your application begins to mold itself around the way your client does business—you begin to deliver packaged solutions that customize themselves!

Penetrating New Markets

You may be faced with a different challenge, trying to enter new application spaces and market niches. You may think twice about attempting to do this because of the increase in the costs required to make this kind of shift. However, CyberGnome-enabled solutions allow you to enter new spaces with more solution capability than the dominant vendors with less development and implementation costs.

Differentiating Your Product

Sometimes your problem is as simply stated, but as difficult to accomplish, as differentiating your product from your competition. Finding that something that sparks your prospect's interest isn't always an easy task. Utilizing new and quite possibly revolutionary technology could very well be that something that would make prospects sit-up and take notice. The CyberGnome, with its ability to learn and gain knowledge, has the potential to be such a revolutionary differentiator. The CyberGnome can help you dramatically differentiate your solution with real value from the current front-runners and do it in a way that gets you a significant window of opportunity over your major competition. It may very well catch people's attention and be the thing that people are talking about.

Tracking the Trends

Staying current with the trends in the computer solution industry is another of those areas that isn't easy—things keep changing, and fast. Just when you thought you'd finished implementing GUI, DBMS, RDBMS, and the Internet, along came the Year 2000 problem. Marshaling your resources to combat that was costly enough, but you know the next trend is already on the radar screen and will eventually impact your business. Many industry analysts believe that the next big wave is Knowledge Management, and some vendors have already hopped on the knowledge management bandwagon, if only by using (and abusing) the term "knowledge management" in their '*market*ecture' and collateral.

Gearing up for knowledge management, or whatever the next big trend is, is a never-ending battle—gaining new skill sets and knowledge as well as changing, testing and modifying product and collateral. Each of these activities increases your costs and impacts your time to market.

With the CyberGnome you can ride the crest of the next trend and force your competition to catch up with you. Delivering intelligent solutions to your clients generates more value at less cost. More importantly, solutions built with the CyberGnome mold themselves to your client's environment, automatically learning site-specific rules and processes. With the CyberGnome you can deliver real value not, just hype!

Decreasing Costs

Everyone is concerned about the ever-increasing cost of doing business. The development and maintenance cost associated with your existing solutions is high. Additional solutions are a major cost consideration—with the additional risk of not getting a return on that investment. Embedding the CyberGnome into existing or new solutions can provide major benefits in development and testing expenses. No programming or knowledge engineering is required regardless of its use in your solutions. The CyberGnome learns what you show it and allow it to experience. It manages the knowledge it gains and can learn continuously. This technology allows you to substantially increase your solution capability without the associated development costs.

What is the CyberGnome?

The CyberGnome is an advanced technological breakthrough in machine learning. The CyberGnome software learns and gains knowledge like we do—by experiencing reality. As it learns, it gains knowledge that it maintains, uses, and constantly refines. Over time it continues to learn and grow in knowledge and intelligence.

The CyberGnome technology mirrors the way the human brain interprets information and learns about its environment—in other words, how knowledge is created and used. Unlike other forms of artificial intelligence, such as neural nets and expert systems, the CyberGnome needs no programmers, knowledge engineers, or AI experts.

Why Use the CyberGnome?

There are many reasons to use the CyberGnome within your solution portfolio:

- It can provide you with the unique and powerful differentiators you need to gain competitive space and marketshare.
- It allows you to build and deliver superior solution functionality to 'raise the bar' in your target markets capturing more business faster.
- It will even allow you to enter major new markets with the ability to effectively challenge the market leaders.
- It demonstrates to your prospects and clients that you are on the leading edge, creating trends not chasing them.
- More importantly, we will continue to enhance and deliver this advanced learning technology bringing major as yet untapped capabilities to the marketplace.

How the CyberGnome Works

The CyberGnome mirrors the way that the human mind interprets information and learns—in other words, how knowledge is created and used.

The CyberGnome is based on the pioneering work of mathematician, philosopher, and logician Charles Sanders Peirce. In the early 20th century Peirce developed a system of logic that attempted to explain the dynamics, logic, and process of learning. Called Semiotics, Peirce's system attempted to explain the process by which we acquire and create new knowledge.

Peirce and Knowledge

Peirce observed that the human mind and nervous system is closed, having no direct connection to our environment. We experience our environment (and gain knowledge about it) by making observations about it. As we observe, we try to make meaning of what we see by mentally comparing it to our existing knowledge.

Through the process of **induction** we determine if an observation or event is valid (whether or not we have previously experienced it). If we have existing knowledge of that specific event we can, through the process of **deduction**, identify the appropriate response and action.

Induction—*reasoning from the particular to the general*

Deduction—*drawing a conclusion by reasoning*

Abduction—*Creating a hypothesis that can be tested*

However, if we never before encountered a specific event we try to understand and reconcile it with our existing knowledge. Peirce proposed that we form a mental hypothesis about the unknown event. He called this process of forming hypothesizes, **abduction**. Abduction creates hypotheses of new and untested knowledge (actually creating new knowledge, which can then be tested through the iterative process of induction, deduction, and abduction. By testing and re-testing, we continue to learn and gain knowledge.

Pendergraft and the CyberGnome

In the early 1960s, Eugene Pendergraft applied Peirce's theories to his own research in mechanized language translation. Gene and a small group of scientists developed a set of theories that became the basis for the work on the CyberGnome. Their work, which continued into the 1990s, was enhanced and has been commercialized by Unisys Corporation.

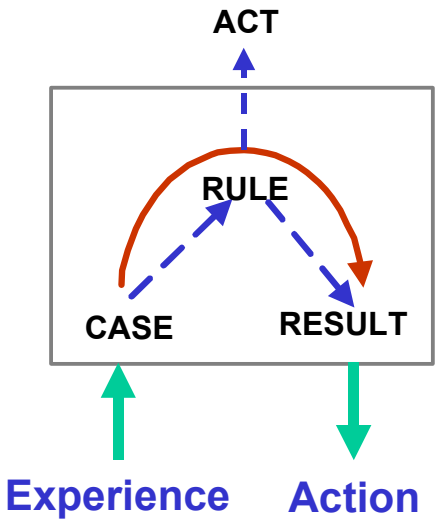
How the CyberGnome Learns

The goal of CyberGnome learning is the discovery and creation of knowledge.

The CyberGnome creates knowledge by acting on its environment, gathering experience, and comparing and validating each experience with its existing knowledge (induction). If the experience is familiar, the CyberGnome identifies appropriate actions and responses (deduction). If the experience is different from its existing knowledge, it forms a hypothesis that it tests (abduction). If the test is successful, the hypothesis is reinforced and added to the CyberGnome's knowledge. If unsuccessful, a new hypothesis is formed. This process continues until existing knowledge is reliable.

CyberGnome Knowledge

CyberGnome knowledge is built and stored as a system of *acts*. Acts are the fundamental unit of CyberGnome knowledge and are comprised of a case, result, and rule.



Creating Knowledge

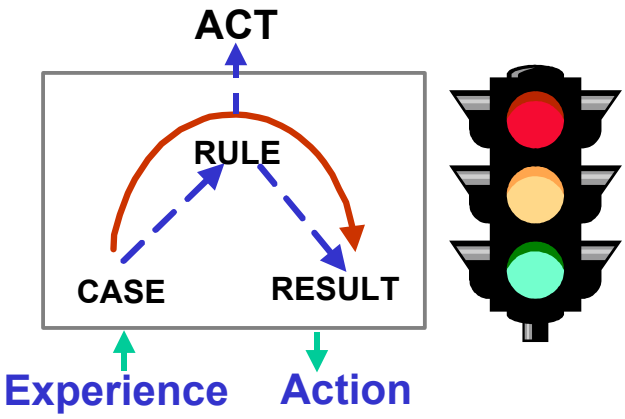
The following explains how CyberGnome knowledge is created and the role of acts in the formation of this knowledge.

Note: The following uses an example to help you understand how knowledge is created. For the purpose of the example, assume that a CyberGnome has been installed in a high-tech automobile of the near future, which is also equipped with sensors that are digitally hooked into its systems. Hundreds of times every second, these sensors monitor the car's environment (condition of the road, objects within view, engine operation) and send this information to the CyberGnome.

How Knowledge is Created

The first thing that must occur before knowledge can be created is that the CyberGnome must experience something in its environment. As the CyberGnome experiences its environment, it perceives unique events or conditions. A specific instance of an event or condition is called a ***Case***. Using our example, as the car approaches an intersection, its sensors perceive that the traffic light is yellow. This information is sent to the CyberGnome, which induces this condition (the yellow light) as a *case*.

From previous experience with traffic lights, the CyberGnome already knows through the process of deduction that red follows yellow. The CyberGnome expresses this information mathematically by forming a *rule* that says, "there is a 100% probability that traffic lights turn red after yellow."

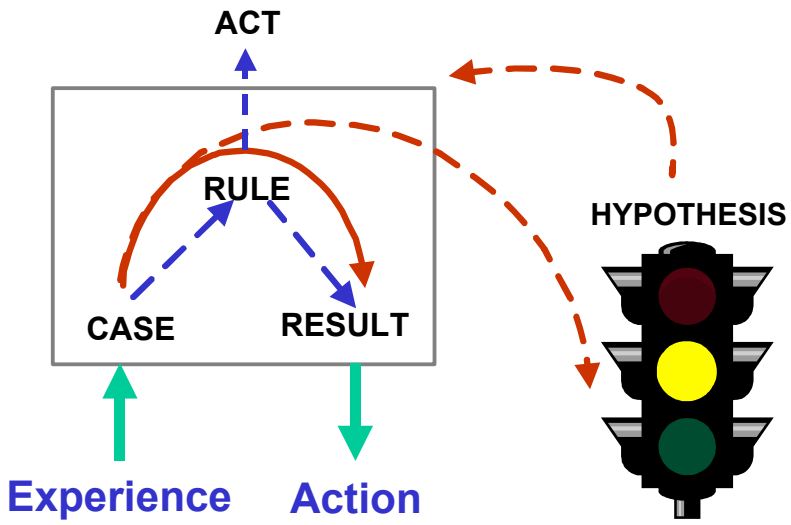


Rules are expressed as the probability that specific cases have specific *results*. Assume that at the next traffic light, the sensor again perceives "yellow". Based on the rule ("there is a 100% chance that traffic lights turn red after yellow"), the CyberGnome will anticipate that the light will next turn red (**Result**).

The CyberGnome will continue to anticipate the result "red" each time it perceives the case "yellow" based on the rule that "there is a 100% chance that traffic lights turn red after yellow". It has learned that for this specific case there is a specific rule and result. And, as the CyberGnome continues to encounter this case/result pair, the rule is reinforced and becomes more and more trustworthy.

Now assume that the car continues on to the next intersection. Here an ordinary traffic light has been set to continuously blink yellow, thus acting as a warning light.

For this instance, the CyberGnome perceives a yellow light followed by another yellow light.



Since this is a different case/result, the CyberGnome identifies (through abduction) a new and different rule, which says for example, "there is a 1% chance that traffic lights turn yellow after being yellow." At the same time it also modifies its existing knowledge by changing the probability of the original rule to "there is a 99% chance that traffic lights turn red after yellow".

In this simple example, the CyberGnome has gained knowledge about only two experiences. But in practice, the CyberGnome perceives large numbers of experiences and builds and tests large numbers of acts. This is how it gains knowledge—or in other words, how it learns.

And, each time the CyberGnome perceives an experience in its environment, it validates the experience against the knowledge it has already learned. By continuously experiencing its environment, the CyberGnome continues to learn, make changes to its knowledge, and become more and more intelligent.

Also, in the same way that the CyberGnome can learn about traffic lights, it can also learn about implicit or explicit processes within your business.

CyberGnome Implementation

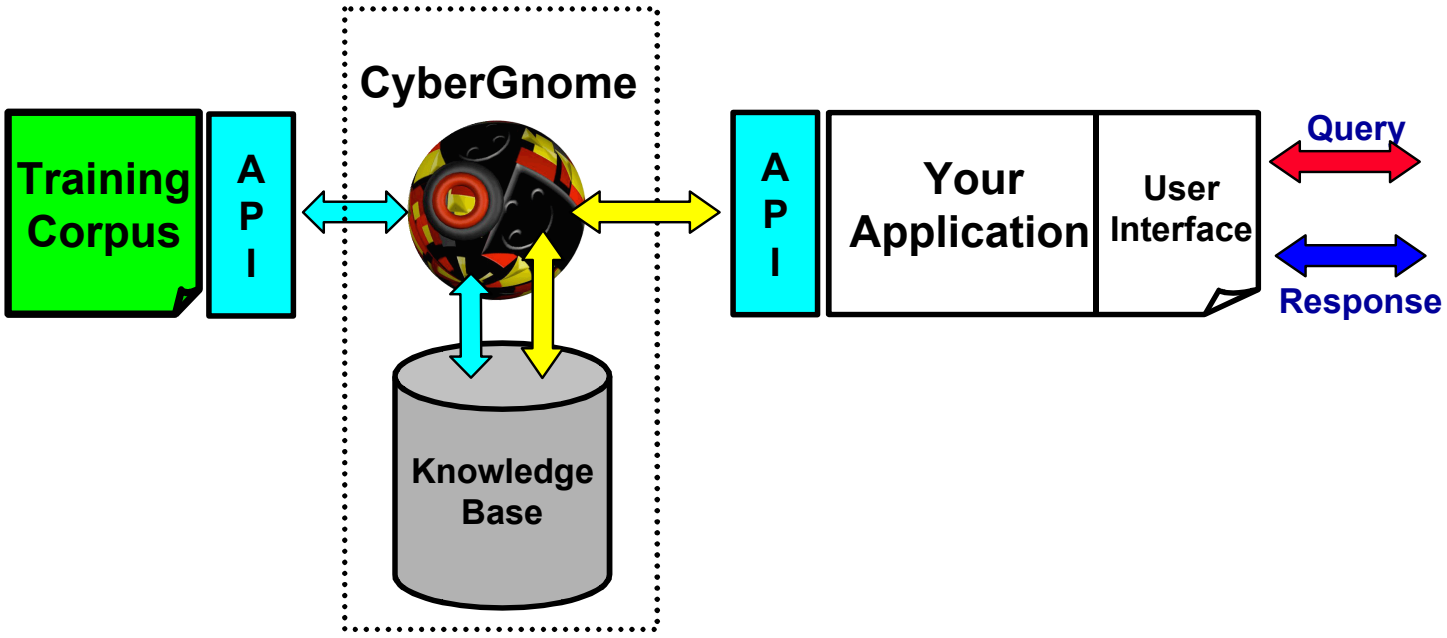
So far this paper has discussed why you will want to use the CyberGnome and also how it learns. This section deals with how to implement and use the CyberGnome within your own software application.

There are three implementation activities:

- Embedding the CyberGnome
- Initial Learning
- Production

The CyberGnome Product

To understand how to implement the CyberGnome technology, you need to understand the components of the CyberGnome product. The following diagram illustrates these components.



A 'Categorization Machine'

The strength of the current version of the CyberGnome is in its ability to learn about categories of objects, ideas, events, and conditions. By pre-defining the answer categories, the CyberGnome is able to learn about the information and suggest to which category a specific bit of information is related. For this reason, we say that this first version of the CyberGnome is a "categorization machine".

The following components are provided with the CyberGnome:

- API—the Application Programming Interfaces are the interface by which different parts of the software access and communicate with one another.
- CyberGnome—the fundamental software code and logic of the CyberGnome
- Knowledge Base—the structure that holds the CyberGnome's knowledge

You provide the following:

- ISV Application—your software, into which the CyberGnome is embedded
- User I/F—the user interface into the CyberGnome. You can either create this or our services group will work with you to construct a user interface that meets your needs.

Embedding the CyberGnome

The first implementation activity is to embed the CyberGnome into your software application.

For the CyberGnome to learn and gain knowledge that is useful to your application, it must be embedded within your software. The CyberGnome's code is fixed—it is the same no matter what application it is embedded in. But, the CyberGnome must be connected to your application in order to learn and gain knowledge.

The following examples illustrate how the CyberGnome might be embedded within a software application.

Example—Help Desk/e-Mail Triage

Assume that your application is used to receive, route, and answer electronic messages (eMail or help desk). Without a CyberGnome, messages come into the system and are routed to support analysts, who answer questions or re-route messages. Now suppose you embed a CyberGnome within this application and allow it to learn about the messages. As the CyberGnome learns and becomes more intelligent, it might provide answers without having to bother a human.

Example—Imaging

Suppose your application is an image scanner, which has difficulty reading forms that have been mishandled. Whenever a damaged form is read, the system alerts a human operator who must identify the form. After embedding a CyberGnome, which learns how to identify each form, it processes damaged forms without having to interrupt a human operator.

Tools to Help Embed the CyberGnome

A Software Developer's Kit (SDK) is provided with the CyberGnome to help embed the technology within your application. The Software Developers Kit (SDK) contains several Application Programming Interfaces (APIs) and a suite of software tools.

The following three APIs are included with the CyberGnome:

- **System API**
  Low level APIs that provide a high performance function call based interface to the CyberGnome's code. The structure and operation of this interface can be compared to the Win32s interface provided by Microsoft to access the system level functions in the Windows environment.
- **Sensor API**
  Sensor APIs are low level I/O routines that interface to input devices. They provide a set of standard function calls for data input to the CyberGnome's code.
- **Effector API**
  Effector APIs are low level I/O routines that interface to output devices. They implement the function calls required to pass the results of the CyberGnome's knowledge processing to the effector modules.

The following modules are also provided as a part of the suite of software tools:

- **Sensor Modules**
  Sensor modules perform the data manipulation and filtering required by a particular software application to provide input to the CyberGnome.

  Presently, there are two sensor modules. An ASCII sensor, which recognizes ASCII characters and a bitmap sensor that passes binary data (bytes) to the CyberGnome.

  As different applications begin to be developed in different domains, we expect that our suite of sensor modules will evolve.

- **Effector Modules**
  Effector modules receive the output data generated by the CyberGnome and process this in a way that is appropriate for your application. This data can then be used by your application to perform a specific action based on the knowledge that was gained.

In addition functions are provided to facilitate communication directly between sensors and effectors.

- **Control Functions**
  These functions control the initialization, setup, and loading of knowledge into the CyberGnome.
- **Maintenance and Monitoring Functions**
  This module is responsible for maintaining the initialization and configuration files.

Initial Learning

The second implementation activity is initial learning. The goal of initial learning is to build the CyberGnome's knowledge and fill its knowledge base.

After the CyberGnome is embedded within your application, the next task is to define and create a *corpus*. A corpus is a body of information from which the CyberGnome will learn and gain its knowledge. For example, a corpus might be an ASCII text file containing all of the HR (human resource) policies of a company, a file of all of the e-mail messages received and sent by a help desk in a given period, or a file of bitmap images.

After the items in the corpus are categorized, the training corpus is attached (through an API) to the CyberGnome and initial learning begins. The corpus is known as a "training corpus" since the CyberGnome is trained using the information with it.

It makes the strange familiar

During initial learning, the CyberGnome learns everything it can about the information in the corpus. It first learns the relationships between the "objects" (for example, letters or ASCII characters). In other words, it learns the "language" of the domain.

It makes the familiar meaningful

Second, it learns the relationships among the objects in the domain (for example, it begins to learn the relationships between words and phrases).

It values the meaningful

And finally, it learns the relationships between the categories (it understands the meaning of a word in context)

Once initial learning is complete, the CyberGnome now has learned what is in the corpus—it has the knowledge it needs and is ready for production.

Testing its Accuracy

From experience we know that different applications require different levels of accuracy. For example, the CyberGnome must be highly accurate when diagnosing medical conditions, but may be less so for less critical applications.

After completing initial learning you will want to test the accuracy of the CyberGnome. Initial learning will show two measurements: accuracy and recall. Accuracy is the number of correct answers the CyberGnome provides and recall is a measurement of the number of questions the CyberGnome attempts.

Assume that after initial learning you find that recall is 75% and accuracy is 90%. This means that *initially*, the CyberGnome is 90% accurate for the 75% of the questions it chooses to answer.

You may or may not be satisfied with these levels, if not, there are two ways to improve them. The first is to change the accuracy and/or recall settings. Decreasing one setting will increase the other. For example, if you want higher accuracy, you may want to decrease recall.

The second and probably most important way to increase accuracy is by enabling the CyberGnome to continue to learn. By providing it feedback and additional information and examples, the CyberGnome continues to learn and become more intelligent—in practice becomes more accurate.

In any case, you will want to test the CyberGnome's accuracy before beginning to use it in production.

Production

The final implementation activity involves the remaining tasks needed to get the CyberGnome ready for live use.

Connecting Sensors and Effectors

The sensor and effector modules must be connected (via the APIs) to your application and to the CyberGnome. As well, the knowledge base must be connected to the CyberGnome.

Going Live

Now you're ready to begin using the CyberGnome in a production environment.

Through the user interface, a query is sent to the CyberGnome. The CyberGnome, having been trained, searches its knowledge base for the correct response to match the query. If it finds a correct response, it sends back an answer or it initiates an action.

If the response that it finds does not fit an appropriate confidence level, a message is sent to the user interface alerting you that the CyberGnome does not feel confident that its answer is within acceptable bounds. You can set CyberGnome parameters to boost or lower the CyberGnome's confidence level thresholds.

Continuous Learning

During production, the knowledge that was gained during initial learning is put to use. If the information in the corpus is fairly static (for example the human resources policies of a company), the CyberGnome's job may be complete. But, more likely, the information changes frequently. Therefore, the CyberGnome needs to continue to learn to keep up with the ever-changing information. The way the CyberGnome does this is through continuous learning, which is the way it becomes more intelligent.

The CyberGnome continuously learns using two methods: feedback and by providing it with a corpus containing additional information and examples.

During production, the CyberGnome may supply an incorrect response because it has not learned enough about the subject and is not confident about its answer. By providing it with feedback about the subject, you enable it to become smarter—in essence, allowing it to continuously learn.

To keep its knowledge current (like a human expert), the CyberGnome needs to be periodically updated when information is added, modified, or deleted. So, the second way the CyberGnome continues to learn is by providing it with a corpus containing new and modified information and examples. By doing this, the CyberGnome continuously improves and perfects its understanding—as it gains more and more knowledge. This is especially important, since the CyberGnome's code does not have to be re-programmed; all that has to change is the information in the corpus.

Depending on how the CyberGnome is implemented within your application, continuous learning may occur interactively (live in response to a question or comment) or it can be done offline in a batch mode environment, each evening, once a week, or whenever is convenient.

But, no matter how it is done, the CyberGnome will continue to learn and become more and more intelligent. However, it will only learn what you want it to learn. If you don't want it to learn something, you don't provide it with that specific information in the corpus or through feedback.

The Power of the CyberGnome

The power of the CyberGnome is that it autonomously learns what you allow it to experience.

The benefits of this are:

- Is not domain specific—the CyberGnome learns about any language, domain, or environment
- Need no modification for use in different types of applications
- Requires no programmers or knowledge engineers
- Continues to learn and improve its knowledge
- Allows you to control the acceptable level of accuracy
- Extends the practical boundaries of traditional applications
- Provides the unique and powerful differentiators you need to gain competitive space and market share.
- Is a new and advanced technology that will be further enhanced to bring major, and as yet untapped capabilities to the marketplace
- Allows you to build and deliver competitively superior solution functionality to 'raise the bar' in your target markets capturing more business faster

References

Dowling, John E. *Creating Mind: How the Brain Works*, W.W. Norton & Company, 1998.

Fischler, Martin A., Firschein, Oscar. *Intelligence: The Eye, the Brain, and the Computer*, Addison-Wesley Publishing Company, 1987.