LINKING BUSINESS STRATEGIES TO INFORMATION TECHNOLOGY - THE CIM CASE

Introductory Comments

I would like to begin with a quote from V.Admiral Tuttle's *Copernicus Architecture* document:

> *Technology without tactical and doctrinal context is merely an engineering curiosity. Operationally, it is a force divider not a force multiplier.*

I want to tell you today how CIM - Corporate Information Management - will become a force multiplier and not a force divider. I shall first describe what is CIM.

The cash outflow of DoD between now and 1997 is $410 billion. Even with force reduction in airplanes, ships and tanks, we are still short of $71 billion of savings. Therefore, Mr. Atwood, the Deputy Secretary, decided that we are going to make that $71 billion difference through effectiveness improvement.

To implement that objective, he created the Defense Management Review Decision process. About half of the projected $71 billion savings are largely dependent on information technology. It is a program that is receiving ample attention. Not only is CIM a Presidential priority program, a Secretary of Defense priority mission, but it also receives personal attention from the Deputy Secretary of Defense. CIM includes now not only looking at the infrastructure of finance, materials, logistics, medical and so forth, but it includes the examination of some of the underlying information flows that support Command and Control. The CIM program office reports to the Assistant Secretary of Defense.

When I came into this job my first task was to find out everything about the needs of the ultimate customer for CIM products. In DoD it became apparent that the place to figure out the customer's critical needs was through the examination of what happens during war-fighting exercises. This involves watching the actions of the "blue" and the "red" commanders. One could learn much about the war business by sitting through post-exercise debriefings and listening to the commanders discuss how and why they planned their actions, what they did, why they did it, and why the "blues" did not succeed despite their numerical superiority.

The brigade commander of the defeated "blues" was equipped with many of

the identical Command and Control means used in World War I. My father was Battalion Commander and his command and control "system" was a topographic map with acetate overlays, and Crayolas to mark up whatever was known about troop positions. That is what I found at our exercises, except that our commanders used radio-phones instead of pigeons and runners. This led me to start looking at the uses of computers in support of the current command and control doctrine.

Let me quote from a recent talk by Admiral Jeremiah, the Vice Chairman of Joint Chiefs of Staff. He said:

Studies of the twenty-first century warfare suggest that the services will have to rely on small stealthy weapon platforms because large ships, tanks and aircraft will become increasingly vulnerable. Low cost, sometimes low tech, responses to our high tech capability suggest that the manned bomber or the large surface ship /may not be relevant/. Start looking at mobile, small, rapidly deployed and jointly managed forces.

Within this framework, I started thinking about the kind of information technologies the DoD should obtain to execute the war-fighting missions before the twenty-first century arrives. The year 2000 is just too far away and much can happen before that. To me eight years is long range thinking. It is too far for an information technologist who must deal with three to four year technology innovation cycles. I explored what could be made available soon to a Joint Task Force commander of a brigade dropping into some forsaken place. I imagined a scenario that calls for only ten days to get in and to get out, to come out with the absolute minimum of losses while protecting property and non-combatants' lives from the destructive power of modern arms.

What we currently have to give to a Joint Task Force commander is just too heavy, slow, expensive, vulnerable and not easily interoperable. This condition is basically driven by our software practices rather than what computers we use. If there is anything that stands out in DoD's current information technology planning is the overemphasis on hardware choices and on hardware acquisition.

Due to the rapid adoption of "open systems" solutions in DoD, hardware choices will become largely a matter of secondary importance. Small scale commercial computers are becoming an inexpensive commodity, because they will be manufactured and sold as mass-produced electronic appliances instead of limited-production scientific equipment. They will be available in retail stores, in blister packs, to be put into shopping baskets. Powerful personal computers will be available for less than $500 within four years - not toys, but 25 megahertz machines. Our soldiers have shown enough imagination to go and get those

devices to support their growing information needs. Our soldiers have already shown during Desert Storm how they can improvise microcomputer-based solutions, independently of the official, yet ponderous, orderly applications acquisition process.

As hardware becomes cheaper and easier to use for solving local needs, allowing the continuation of improvised solutions will only magnify the chaos arising from a lack of interoperability among independently conceived solutions. The solution is not in imposing an even more demanding oversight process – which means another bureaucratic obstacle – on the imagination and adaptability of our people. We must break the current stalemate of how software is acquired and integrated. We must devise a systems design "constitution" that will make possible local choices within the limits of rules that are essential for the preservation of easy application interoperability in the battlefield.

DoD has currently about 1.4 billion lines of code. Most of it is not adding any administrative value or contributing to our Defense strengths. Yet we can't eliminate the unnecessary code because it is performing tasks that are essential for passing data back and forth among incompatible and inconsistent applications. What's worse, we are continually adding to non value-added systems thus magnifying the economic damage to our ability to fund much needed improvements.

One of the primary technical purposes of Corporate Information Management is to bring order into the pervasive software chaos. Our objective is to bring the software and applications environment into an orderly framework that will allow the field commander to receive personally relevant information – often from the home bases in the US – as needed so that our troops can plan and execute missions with more complete and timely information than is currently the case.

How will that ambitious goal be accomplished? Let me recite selected principles from a forthcoming report which will be most likely titled *Managing Defense Information* .

As the first principle, let's treat personal computers as any other supply item.

Make Personal Computers an Item of Supply

Personal computers now costs less than an artillery round. For instance, a shell for an M1 tank costs $1,600. Rocket-propelled ordnance costs ten to fifty times more. The most prevalent hardware that I see in the armed forces of the

future is not a rack-mounted or desk-top computer as we know it today but a device that looks very much like a clipboard. These are already widely used by truckers, delivery services and the police. These are sturdy devices, with liquid crystal displays, without a keyboard, without removable magnetic disks because they are electronically connected by means of a wireless LAN to central data sources. The screens can provide accurate displays of topographic and combat intelligence. The commander interacts with this "electronic clip-board" by means of a stylus. He can query a topographic map for relevant information.

The battlefield commander's workstation looks very much like the prototype I saw recently in actual operation. This innovative combat unit has compressed their brigade level Command and Control system down to a five pound lap-top computer for field deployment. It has real-time displays of positions not only of our own forces, but also relevant information about weather, detected enemy movements and the movement of other friendly forces, in the air, sea and on land. On this device you can zoom in and zoom out for low level details about supplies, manpower, readiness and plans. This experimental Command and Control system operates in an identical manner whether in a planning, mission simulation or mission execution role. This means that a soldier's training – the single most expensive and valuable asset in DoD – is continually enhanced through learning how to use the same device.

The prototype portable Command and Control computer now in the hands of the commanders costs only about $3,000. It is connected to minicomputers that provide communications and database services. The prices of commercial palm-computers, clip-board computers, lap-top computers, desk-top and minicomputers are now declining at a rate of 35% per year. In comparison with the prices of other battlefield assets these cost are ceasing to be decisive. As other DoD costs escalate, our commanders will quickly figure out that favoring improved information capabilities will be their preferred solution in coping with reduced budgets. Unfortunately, such attractive tradeoffs are not easily available right now.

One of the reasons why our existing deployable Command and Control devices still cost ten to a hundred times more than commercially available devices is because we are buying computers on the presumption that it must be capable of not only displaying a wide range of data, but must be also capable of managing its software. We intend to change that.

DoD has now defined the characteristics of its software tools (I-CASE). The underlying idea of this approach is to sever the tight coupling that currently exists between software development computers and "run-time" computers. We shall

provide powerful workstations to our software developers, including the means for automatic generation of run-time software. Our intent is not to rely forever on procedural languages such as Ada, COBOL or JOVIAL. The big productivity gains in software will be realized by re-using standard software "objects", e.g. software components that are defined by high level requirement statements. These "objects" can be transformed to fit the unique characteristics of the machines that are actually in use by the soldiers. This fitting is accomplished by the toolset that transforms software from the DoD standard development "repositories" into code that fits into of cheaper machines. In this way you can reuse the same software on a variety of inexpensive machines that run powerful programs, without most of the difficulties nowadays imposed by separate procurements that fit local systems solutions.

Let me repeat: one of the CIM principles is to make personal computers a field issue item and not something that is treated through a completely different process than any other weapon or supply item. The idea here is to provide the field commanders with superior information management means, wherever and whenever required. As a matter of fact, there are special operations missions that I can readily visualize, where every member of the team would have a personal Command and Control compute/display/communicate device to coordinate actions with ballet-like precision.

This brings me to the second CIM principle. Let's impose a well organized systems integration discipline on all existing and future DoD information systems.

Apply Systems Integration Disciplines

In order to allow a widespread distribution of Command and Control to commodity hardware we must be able to integrate the information flows among existing computer centers that house databases. We must be able to share messages without any difficulties between every communication source. We must assure that information can pass from one application to another. Our current information assets (e.g. legacy hardware, software and communications systems) can be best described as isolated automation fortresses. Unfortunately, many will be with us for decades to come because we cannot afford to convert them immediatly. The key to CIM is not how to design new "ideal" systems solutions that will instantly replace everything that we own. CIM must chart a course that will allow us to gradually migrate from existing solutions, in carefully planned evolutionary steps, towards broadly defined long-term objectives.

Our primary economic objective is to reduce the operating costs of the

current information establishment by at least six billion by 1997. Simultaneously, our design objective is to make it possible to support rapid deployment of Joint Task Forces with superior low-cost, survivable, easy-to use, inexpensive computers. We must do all of this while not breaking the existing wild diversity of applications that were conceived to serve narrowly defined DoD functions. The only way how I know how to do this balancing act is through strengthening DOD's information integration capabilities. In the age of information warfare and of information-based combat advantages systems integration becomes one of the core competencies of the US fighting forces.

From the standpoint of the rapid deployment needs of a Joint Task Force commander who needs a one-of-a-kind planning and execution control command post, DoD does not presently possess an adequate systems integration capability. We must own a pervasive and all-encompassing systems integration structure if we ever are to achieve low-cost and responsive systems interoperability.

In DoD everybody claims to be a systems integrator – for their own isolated program or for their unique application. The difficulty lies in that there are as many systems integrators in DoD as there are acquisition contracts or local area networks. That's a large number, perhaps in the thousands. You could not get all of the DoD system integrators into a good size ballroom. Since all of them tend to their own mission and technical inheritances they have legitimate reasons why they cannot possibly consent to a common technical solution. DoD is perhaps the world's most diversified organization. The idea of solving the problems of integration by means of forceful imposition of absolute uniformity is unrealistic. To cope with the diversity and rapidly changing needs of DoD, we are proceeding with an approach to systems integration that embodies the principles of a distributed and delegated "constitution" for making functional and technological decisions.

For instance, there shall be certain reserved rights at the OSD level, which we now designate as "enterprise level" integration. At this level, we prescribe data management and software asset management. This is not theory, but something we are placing into effect. As an example, we already have in place both the policy and the execution mechanism for a DoD data repository. It will ultimately include all shared data definitions. There will be no exclusions. The data repository is an enterprise property. Services will not "own" data any more, unless the information is totally service-specific. There will be no Air Force data, no Navy data, no Finance data, no Personnel data if any of this data must cross service or functional boundaries. To achieve coordination of Joint missions, "enterprise" level data will be held at the OSD level. One of the priority goals of CIM is to see to it that the DoD data management program is firmly

institutionalized within two years.

There are other "enterprise" level standards, such as in data communications and in data center operations, that are presently defined and maintained as service-specific solutions. We are in the process of organizing a DoD information "utility" that will assure communications compatibility from keyboard to keyboard. For security and systems integration reasons, we must have the capacity to have total network visibility throughout DoD. The current proliferation of individual networks, perhaps as many as one hundred and twenty-five separate data networks, must be brought under central management. I don't know how many networks there are, but we are going to have an "enterprise" level structure under the management of the Defense Information Systems Agency because there is no other way how we can assure full communications interoperability. Communications is the lifeblood of all information management and there can be no partial solutions to that.

"Enterprise" level integration also calls for the creation of a computer services utility which will provide the back bone computing structure for DoD. We do not want our commanders to have to specify where their computations or data base services come from. The soldier in the battlefield with a $500 display should not give a hoot where his or her data are retrieved from. In future warfare, it will be our information networks and data centers that will become the choice targets for any enemy, particularly fifteen year old kids, running around with rocket propelled grenades. Therefore, data and communications services must be shared and distributed. We must field information source redundancy, which means to me a widely distributed shared information "utility" where information power can originate from many sources. By redundancy I don't mean two and I don't mean three alternative sources of computing and communications power. I mean five or more. Therefore, the DoD computing nodes and the databases must be totally redundant and location nonspecific. In fact, they must be able to move around like a giant shell game. Enterprise level integration puts into place an environment so that the field commander can focus on the task at hand which is mission planning, mission execution, and not have to worry about the location or compatibility of the computer and communications links.

At the enterprise level, we are also specifying a new business process redesign method. The current approach to computer systems acquisition is patterned around the life cycle management process that may be suitable to the acquisition of carriers or bombers, where everything is defined – in complete detail – at the point of completion of the design. Information systems cannot be designed that way. Information systems designs must be driven by the ultimate customers, such as field commanders or light-line maintenance mechanics. We

have heard of the doctrine that we "train as we fight". My approach to information system design is that we "design as we train". Design as you train means that you give your field commanders the capacity to rapidly customize systems design within the constraints of an "enterprise", "mission", "function" and "applications" architecture that is defined by the DoD systems integration "constitution". We have now defined what is reserved as "enterprise" integration tasks – with everything else delegated to lower levels of integration. We are now in the process of proceeding with defining the additional three layers of systems integration. The ultimate objective, however, is to make it possible to make information systems serve the unpredictable and unique needs of individuals.

This brings me to the third CIM principle. Let's use information technologies as a means for increasing the ability of people to use their initiative and to exercise greater freedom of choice.

Increasing the Customer's Capabilities

For instance, our objective should be to make it possible for a commander to go on a training exercise and within twenty-four hours after debriefing, to get a new software release that corrects for the latest deficiencies. With a standard systems architecture, integrated computer-aided software engineering and a software "objects" repository such capability is already commercially available. The beauty of this approach is that for the first time you could start looking at computers as truly "personal" devices. What you buy today, and call a personal computer, is not. The appliance on your desktop today is a hostile beast that doesn't even know your name. A true "personal" computer is not a piece of hardware. It is software that recognizes your individual habits and tries to compensate for your information-handling limitations. It is your shadow. In fact, it is an extension of your thinking processes.

Commanders are different. Commanders acquire their view of the battle space and their biases during throughout their careers. They should be able to carry their thinking "templates", similar in structure to football plays, from computer station to computer station as an extension of their experiences. The new technology of "object" oriented programming lends itself precisely to this approach. I call the portability of thinking patterns into the battlefield the "shadow warrior". A soldier should be able to call on thousands of "shadows". These would not only be a reflection of his own background, but also the experiences of those who have "walked" this way before. In the confusion of the battlefield, when the information load is perhaps the most demanding in anything a human may ever experience, these "shadows" can assist in coping with situations to which a successful response pattern has been found.

In the current practice, when you go to a War College, when you play war games, when you train, you are supposed to learn something and remember it for future use. As dependence on computers as a training and simulation device grows, much of this experience is wasted because when you leave a war game, there is nothing for you to carry with you except what you may remember. Under the pressure of the battlefield when you have hardly slept for forty-eight hours, there is a lot of heat and smoke and your senses are overloaded with radio chatter, you cannot recall very much. We want our commanders in the field to be well composed to be able to think deliberately. That means that we have to reduce their current cognitive information overload. During a recent exercise at the National Training Center I listened in on the radio talk. When it finally got down to the critical moments during an assault, there was no way how any human could have fully understood all of the messages. Most of the chatter was just position information. It got worse when the tank commanders had to close their hatches because of a simulated a chemical attack. They could see the hostile world around them only through a small aperture. They became totally dependent on the radio for cues about the overall situation.

The best way how to reduce the information overload in the battlefield is to shift as many of the audio messages to the visual form. The battlefield of the future must therefore become much more digital, so that graphics can carry most of the routine positions and status information and be available on request. This does not mean that you do not resort to voice communications when that is necessary. Voice is the ultimate message for conveying stress. But for position information, for analytic information, for situation assessment information, for interoperability and for comprehending the context of the local action you must go digital. You must go digital because the ability to communicate in pictures greatly relieves the information overload on the scarce, vulnerable and insecure wireless channels.

In the future we must provide a wide range of information easily, in color, in real-time and in simulation mode – if necessary. The system must be able to process rapidly unplanned inquiries to display temporarily related data, such as troop readiness, ammunition supply, enemy positions, intelligence estimates, weather, personal information about soldiers or what have you. The battlefield has to be digital. It is regrettable to see that my Federal Express delivery person and my local policeman have more computer technology in their vehicles than our ambulance or ammunition truck drivers. The readiness and deployability of our forces will be greatly enhanced by providing most of our soldiers with graphical and disposable digital information management devices.

This brings me to the fourth CIM principle. Let's use information

technologies as a means for increasing the ability of people to use their initiative and to exercise greater freedom of choice.

Preservation of Accumulated Training Skills and Security

The most valuable asset – based on the estimated acquisition cost of trillions of dollars – is the accumulated know-how of the DoD's fighting team. As DoD shifts its decision-support practices towards automated systems, an ever increasing share of operating expense will have to cover training how to use computers. Over the life of any computer system, the training and error correction costs will always exceed the initial development expense, unless we learn how to share training across a wide range of systems. Therefore, one of the CIM objectives is to conserve accumulated training skills.

How do we do accomplish this? How do we preserve "shadow warriors" from generation to generation of computer hardware as it is replaced every five to eight years because of obsolescence?

This brings us to the "Standard InfoWindow" (SIW). Training involves how a soldier reacts to a display. The software then triggers the long chain of logic that lies between the display and the software "object" library. Everything in this long chain, such as compilers, operating systems and hardware must be transferable as one migrates from one technology solution to another. The training cost is captured not in software, but in the man-machine interface, what the soldier sees. The cost of this training goes up if we have a confusing interface that changes from application to application. Not only is this costly, but such variability is error prone, which induces additional cost for checking and for monitoring.

Recently I went to visit a key command center and saw in front of our commanding general's station four computer displays. I looked at the keyboards and there were five by seven cards taped to the shelf, with reminders that on this machine "Control/B" means this, and on another machine "Alt/F1" means the identical command. We cannot tolerate such variety because it is dangerous. Therefore, we are proceeding to adopt a standardized display style, "Standard InfoWindow" (SIW).

Some of our command and control systems have already done a great deal of screen interface standardization as a way of simplifying training needs. So, there must be something good about standardizing what the troops get to see. We must be able to present a standard view of all applications because you never know who will end up at what battle station, at what time. Troops don't have time to take a two day course to learn what each unique screen symbol represents

if you may have only minutes to start acting. We must create an easily comprehended computer environment so that people will be able to plug into their personal computers their "shadow warriors".

Standardization of the computer interfaces is also necessary for getting answers to unique queries. This is the concept of a specialized "shadow warrior" now designated as an inquiry "agent". You just enter, or speak – if you have voice input – get me this information. This presumes that our military networks will be sufficiently standard and intelligent to recognize the characteristics of one-of-a-kind "agent" and automatically route them to "servers" that can have the answers. The idea of a location-independent software "agent" is necessary for survival whenever individual network elements become disabled.

The future network capabilities will not be as easy to get as you may think based on the ideal descriptions of "shadows" and "agents". How do we safeguard information if we allow thousands of software "agents" to rummage through our electronic files looking for information? What are the tradeoffs between rapid-response and protection against unauthorized intrusion? How do we prevent destruction of our files by viruses and software "bombs"? The current approaches that impose elaborate security procedures also impose cumbersome delays.

We must put in place networks that allow a large share of inquiries to be handled automatically without clearance for each event. We must have systems that will rigorously authenticate whether you are allowed to gain access automatically to a file. If you have not been adequately pre-authorized, the network will route you to a human gate keeper to screen your information. Can we expect to get fully automated multi-level secure information systems in the near future? I do not think so. For information security our command systems will have to rely on human gate keepers. But instead of "sneaker net" security, in which secure information is physically transported from one network to another we will depend on compartmented workstations where a gate keeper can see the incoming software "agent", validate it, and then allow its the passage to the next checkpoint.

But training and security are not the only critical assets that need preservation as a matter of highest priority. The DoD of the 21st century must be able to guard the accumulation of its programmed knowledge. The software "objects" and software codes will contain much of the institutional know-how for integrating the various components for predictable results.

This brings me to the fifth CIM principle. Let's use information technologies as a means for increasing the ability of people to use their initiative and to

exercise greater freedom of choice.

Preservation of Programmed Knowledge

We are pursuing a policy of conservation of accumulated software knowledge by specifying computer-aided software engineering tools. This toolset will convert software objects residing in the DoD software repository into machine-specific computer programs. Our objective is to achieve an "Open Systems" architecture independent of the fads and fashions of various "Open Systems" claims. Right now there are too many versions of UNIX that claim to be "open". What is recognized as "open" today may not be so ten to twenty years hence. The choice of ADA as the DoD procedural language of choice seems to be right at this moment, but may not stand up twenty to thirty years from now if the commercial computer industry adopts object-oriented dynamic languages to write industry standard software components.

We must retain the underlying software knowledge from one technology generation to another. We must be able to build and evolve on top of what we already own instead of scrapping it every time there is a major shift in the economics of computerization. Unlimited software portability through adoption of some simple "Open Systems" standards, such as POSIX and GOSIP, are just another wishful figment that is fashionable at present. Unless we instrument our software production environment with a standard toolset we cannot hope to achieve reasonably affordable transport of our software from one computer to another.

Just buying a UNIX machine and coding it in Ada, will not adequately safeguard the accumulation of software in this form. POSIX and GOSIP compliance will not assure the portability of personnel information from mainframes to laptop computers. This is why the I-CASE toolset that we are going to buy will become the ultimate guarantor of software interoperability and portability, in the same way as it makes no difference to a screw driver whether you have made the screws by casting them, milling them, punching them or forging them, or whether you have them out of steel, galvanized iron, or nickel. DoD must equip its developers with a standard toolset that will shape how a the particular software implementation is manufactured to fit into a standard repository of programmed knowledge.

We are simultaneously proceeding with a vigorous program of standardizing application interfaces that will allow re-engineering software from implementation to implementation. From the standpoint of interoperability, we expect to be able to take standard software components produced by an

aggressive and imaginative commercial software industry and include their products into the DoD software inventory. Right now, DoD software is hand-crafted by contractors as unique artifacts. This software is custom-fitted into one-of-a-kind hardware environments, with no chance of reuse, even for identical purposes. This process is very costly in the same way as all artwork costs more than a mass-produced product. We have to cease looking at our software acquisitions as the commissioning of a sculpture or of a painting. The construction of software must start by treating it as a controlled, predictable manufactured process that uses mostly standard components that we can buy anywhere in the world prior to integration to fit a specific DoD requirement. The skill and artistry in the future will then come not from the making of components, but from the way they are put together.

What are then the technical directions for satisfying our future software needs? We must adhere to the policy of pursuing only evolutionary and incremental systems acquisition. There is no way of designing an information system by first writing detailed specifications which define every screen that people will be looking at in 1999 or in years beyond. The current acquisition process is faulty in that it attempts to define with certitude that which is unpredictable. It is physically, logically and rationally unfeasible to specify in any great detail a command and control system or business application for everyone, under all conditions for the next five to ten years. Nobody can do it because it assumes the existence of a perfectly planned environment. So why even try? The current acquisition process dictates the production of all-inclusive specifications on the presumption that information systems are a completely definable clock-work artifact. They are not clock-work like. Increasingly, information systems are taking on the characteristics of what they support – people, organizations and politics. Information systems are as hard to define and to manage as any other social organism. Evolutionary learning and the balancing countervailing demands by means of a generally agreed "constitutional" process is the American way of managing evolutionary complexity. Let us apply this method to how we manage communications under rapidly changing circumstances.

The best you can do is to specify the directions how you will gradually evolve from what you already have to where you may end up someday. The Assistant Secretary of Defense for C3I has just promulgated a revised evolutionary acquisition life cycle management system that encourages DoD to proceed through evolutionary and incremental system deployment based on gradual acceptance of innovation by customers. The objective of the new process is to relieve the burden of having to commit too soon to a detailed plan that nobody believes anyway, except perhaps some auditors who will hold you

accountable for what an anonymous writer stated more than a decade ago.

Summary

The objective of Corporate Information Management is not to promote information technologies purely as a an abstract exercise that will come to be forgotten as many other similar ambitious schemes in the past. I want to convince you that we are thinking in terms of the kind of cost reduction and technology plans that DoD must have to cope with war contingencies that may reappear any day.

Thank you very much.

Questions and Answers

Question: There has been a lot of talk about the Department of Defense industrial base. In a recent symposium, the question was posed of Dr. Reis as to the future of Ada and how it contributes to the nation's industrial base, since it is only mandated by the Department of Defense and used by no other government agency, except for the DoD, nor to my knowledge in industry. My question to you is what do you see as the future of Ada within the DoD?

Answer: It is not a correct statement that Ada is only used by the Department of Defense. Ada is being used commercially, although not extensively. When I came to DoD I had strong arguments and strong biases against Ada. Since I never used Ada at Kraft, General Foods or Xerox, it could not be any good. I went to see "Mr. Ada", Lloyd Mosemann, and said to him "We are going to have a shoot-off between other procedural languages and Ada to see how Ada compares." We commissioned three totally separate groups to investigate the merits of Ada. The question was that if we do not pick Ada, than what? COBOL? The answer of the community was, we don't want COBOL. FORTRAN surely not. forget JOVIAL and PASCAL. PL/1, no way. The software people than started raging for C++. So that's how we authorized having a comparison between Ada and C++. The separate teams found that last year Ada was still superior, much more mature on the progression of software languages

than C++. That was a unanimous conclusion.

Now is Ada the ultimate solution? I do not think so. Ada is the best of all the bad choices we could have made. Ada makes an awful lot of sense, particularly for real-time systems, but most importantly, the Ada is a necessary stepping stone towards an object-oriented environment where we ultimately wish to end up. Will DoD code in procedural language called Ada fifty, forty or thirty years from now? Mostly likely not. We will be coding in a much higher requirements level language which will then execute into Ada or whatever is most the most efficient choice for a particular situation.

So to answer your question, Ada does have a future. For DoD Ada today is the best of all the other choices. I have no doubt about that, especially since I am not getting any requests for waivers from Ada.

Question: What sort of efforts are going to be made to salvage a lot of this /existing/ code? Is there going to be a massive re-engineering effort or what are your projections for updating that code?

Answer: I have a specific example right now, and I am going to use a trivial example just to dramatize it. So far as we know, we have more than 50 payroll systems in DoD, and I am sure there are more than that. We are in the final process of selecting the software environment for the payroll system of the future which will be tightly coupled to the personnel information system as well. We will institute a re-engineering effort for the best payroll system we have already and then discontinue everything else.

DoD does not need 1.4 billion lines of code. Most of the code that we have is redundant. We do not need it. Let me give you an example. DoD must have thousands of lines of unique code just for generating a "date". If you want to generate date through an automatic clocking routine, it will take anywhere from 2,000 to 4,000 lines of code to do that well. Individual "date" generating software is now maintained at hundreds of sites. In fact, our contractors sell and resell us a slightly modified "date" generating code over and over again. Most likely we are going to have two "date" objects in DoD. They are now in the standard software library, called RAPID. It is going to be issued as government furnished equipment. We do not expect our contractors to write "date" routines any more. There is much code we will not even try to re-engineer. We are just going to eliminate what is not needed any more.

Question: I am involved with the Air Force Mission Support System program, which has many of the characteristics of what you have talked about in terms of evolutionary acquisition incremental deployment, and there are some things happening at the program level that are similar to some of the things that you have talked about. I am wondering is this something that you are doing from the top down and trying to drive it down to the program level, or are you actually working at some program levels also?

Answer: You are asking about management style. My approach is first to try something out without policy because DoD is a place where policy does not necessarily result in implementation. You have to first try and find what works. Every technology that we would like to ideally have already exists somewhere in DoD. We have a formal mechanism for publicly recognizing what should be emulated by others. It is called a "Gold Nugget" award. At present I find little need for original invention. My usual approach is to find a place where things work. Now, they may not work perfectly. Usually, such activities are under-funded. Usually, they operate more like a guerrilla action team than a well structured command. We look at it, then we give it little money, and then start building consensus and acceptance around it. That is what is called bottom up approach. Once you have done that and it is a success, then you come in with high flying plans and policy. In CIM we also have a policy shop that is cranking out DoD directives and guidelines. But usually that comes six to eight months after discovering a Gold Nugget.

I do not believe that things work only from the top down. Only after you have found something that works, will policy and ample money be effective in making the innovation an integral part of the DoD institution.

Question: One of the major issues facing this group is how multilevel security compartmented mode or systems security will help the unit planning system interoperate, not only within its own unit structure, but also with various echelons. Would you speak to the evolution of security policy?

Answer: On my first day at DoD, I tackled Ada. On the second day, I pursued MLS /multilevel security/, because everyone told me that my industrial background is irrelevant in view of DoD's stringent security requirements.

So I went for a three hour briefing about MLS. They told me everything about what cannot be done. One of the wonderful things about MLS in DoD is that there are so many attempted approaches, but without any acceptance.

Help is on the way. We now have an outstanding new Deputy Assistant

Secretary of Defense for counter-intelligence who hired an experienced Director of information security. They are now proceeding with the implementation not of a perfect MLS system that deals with all of the exposures that can conceivably exist from now on. Their approach is to implement what is feasible and what makes practically sense, which may, however, require some human intervention. There is now a candidate software solution to MLS. It is being tested now. In fact, it is running, and I just want to know when it can receive a Gold Nugget.

Will the approved MLS solution be perfect? Most likely not. Will it be totally automated? I do not think so. But it will work, and it will provide an answer to many of the practical security issues that are presently inhibiting further progress on a number of applications.