RECORD, Volume 24, No. 3[*]

New York Annual Meeting
October 18–21, 1998

Session 3PD
A Sure Way to Increase Efficiency: Make Insurance Systems "Speak to Each Other"

Track: Computer Science
Key Words: Computer Systems, Product Development

Moderator: MICHELLE D. SMITH
Panelists: CHERYL A. KRUEGER
MARK ORLANDI[†]
STEVE RYDZEWSKI[‡]
Recorder: DARIN G. ZIMMERMAN

Summary: The panel proposes solutions to the problem of having many different data storage and analysis systems that don't "speak to each other." Systems such as valuation, administration, modeling and projection, accounting, and asset systems are often from different vendors on different platforms. Panelists discuss data, software, and interface standards as methods of improving information shared among these systems. Standards developed for other areas of the insurance industry and other financial service industries are discussed to illustrate these concepts.

Ms. Cheryl A. Krueger: I'm going to talk about the benefits of data integration. I would like to describe the current environment that we're working in based on a product perspective. I'll then share with you the results of a survey that we did to show how automation can help to make your company work more effectively. Next I'll talk about the opportunities that exist when data and systems are integrated.

I will start with a product focus. You can look at the data integration problem in different ways, but products are really the driving force behind what we do. We start with idea generation somewhere in the marketing or product development

†Mr._Orlandi,_not_a_member_of_the_sponsoring_organizations,_is_Director_of_Company_Services,_Standards_ Department,_of_ACORD_in_Pearl_River,_NJ._

‡Mr._Rydzewski,_not_a_member_of_the_sponsoring_organizations,_is_Vice_President_of_Software_Development_at_ PolySystems_Incorporated_in_Chicago,_IL._

area. Then we move on to some pricing work. We then get into implementation and sales. Finally, we get into the area where a lot of us work, that being the monitoring of the results of what's happening to those products. Throughout that whole process, we use all types of different systems to do the work required by those different steps.

Our ideas turn into product parameters and features that need to be input into a pricing system. Through pricing, we develop commission rates, premium rates, and so forth. These rates need to be entered into the following systems: administration, illustration, projection, reserve, and any other relevant systems. The data sharing continues throughout the whole process.

Let's go through a few of the inputs used in this process. We start with product features as the major driver. The plan definition has to go into the pricing system, the administration system, the reserve system, the projection system, and the illustration system. When we're doing our pricing, we get our assumptions from data that we have in-house or from outside the company. That has to go into our pricing, projection, and reserve systems (for GAAP reserves). We have to provide data to our reinsurers and we have to set up our agent compensation system. You can see where I'm going with all these data. If you display it graphically, it's not a pretty picture in terms of how much data needs to be shared between these systems.

How did we, as an industry, get here? Many of the systems that we have were developed to do what they do best, but not necessarily by the same people. Vendors developed some systems. Some of the systems were developed in-house. As these systems were developed, there wasn't a whole lot of push from within to get multiple systems to talk to each other. In the past few years, it has become difficult for the every day user of the data to retrieve this information from the mainframe system. We became a culture that required a programmer to get access to the data we need. That's not necessarily true anymore, but historically we're used to going through some trouble to get our data.

Some vendors created their own subsystems rather than coordinating with other vendors. Thus, if a software vendor had a really good administration system, rather than work with a valuation system vendor to move data between the two, he or she might have created something that would calculate reserves—possibly for some of the products, maybe for all of them. Even if the vendor did an excellent job in creating this new information, it was hard to get the data in the format you wanted.

The demands that were placed on systems were adjusted to fit within desired time frames. People weren't demanding five days to close the books. We didn't have to monitor weekly lapses like we now do on some lines of business. Nobody demanded quarterly plan updates because it took an entire quarter to develop the plan.

What was the impact of this environment on our productivity? We surveyed some companies. We wanted to find out how long it takes from the time the actuarial area gets the data they need to close their monthly or quarterly books to the time they passed the data on to someone else, and then to move those data through all the different manipulations needed to get it from their administration system into their reserve system and so forth. We had these companies grade themselves based on how highly automated they thought they were by choosing one of four different levels: (1) highly automated, (2) moderately highly automated, (3) moderately automated, and (4) not very well automated. Two-thirds of the companies graded themselves as highly automated or moderately highly automated. Those companies had an average of six days for their periodic processing compared to 11 days for the companies who said they weren't very well automated.

Assuming that you close books monthly, you can automate this process to the point where you could add 60 days of productivity to your department. With the shorter time-frame you're not managing data, you're actually doing analysis or can add some value to the company. That's why data integration is so important. I'll go through a couple of examples of things that we've seen in projects where we've tried to increase the efficiency of some area of the actuarial function.

First, let's look at the monthly reserve calculation. The first thing to do is summarize the administration data in a format that goes into the valuation system. The raw data comes from some other area of the company. You have to transfer it to a PC and reformat it for your valuation system. Then you run your valuation and export the output to a spreadsheet. Next, you run some macros to create your management memo and your general ledger entries. You send out a few pieces of paper to the planning area, and they go through their actual-to-expected analysis. Then they send some pieces of paper to the financial area so they can do their general entries. We mapped out this process on a month-to-month basis, and looked for areas where we could increase the level of automation.

The first idea we came up with was a revised process where we first provided data in the correct format to the valuation system. We found that people don't systematically look for opportunities to cut down the amount of data transfer when they have to manually intervene with the system. After the data is transferred to the PC, the valuation is run. We integrate the reserve output into an actual-to-expected analysis and to the general ledger automatically. We automatically reformat the data from the reserve system to be passed into another system and into another area of the company. For example, if we have to send the actual results over to the planning area, we integrate the actual results in with the expected results.

The second idea is to take the reserve output and automatically update the general ledger. Looking at this monthly valuation process, we can gain efficiency by integrating the outputs from one system with inputs to another system.

The second example of data integration potential involves systems that share the same data. For example, when bringing a new product on board, we finalize the plan specifications and distribute them. What we often see is a product book that describes what the product does. Everything is defined, down to the ones and zeros in the administration system turning different features on and off. This product book gets distributed to all the implementation areas. The administration system gets updated with the plan specifications that it needs to run. That same data, but in a different format, go into the reserve system because it is updated by someone else. The illustration system needs another format to run their product illustrations, and finally the projection system gets updated. Sometimes there's some overlap with the pricing and projection system to make that a little bit easier.

What if we could have data in the same format to create the plan specifications and assumptions, and be able to share those between these systems? At this point, such sharing isn't done very much at all. Many of these systems are vendor systems, and we haven't seen a lot of push to have the vendors dedicate resources to put such a process together to update systems automatically.

How do we solve this data integration problem? One idea would be just to have one big system. I don't think this is a feasible solution. Another way to try to integrate data is to create a sharing structure within each company. The company has picked different vendor software to work with, and they have their own in-house software that works well for some things. Any company could dedicate some resources to do data sharing and actually have somebody go to the board and map out where they can transfer data efficiently to improve what they're doing. When we're looking within the company, where do we have our biggest holes? Where is the most time spent transferring data back and forth? Then we can find where we need to dedicate time to cleaning up that process. A data/work flow chart can open your eyes to some opportunities. One concern is working with vendor systems. You rely on the vendor to keep either their data input or their data output in the same format. You may have some difficulties when the vendor comes up with an upgrade to their system and hasn't considered the work-arounds that you've put in place for data sharing.

Another potential solution is data standards. This is already working in a lot of industries. I realize we have a rather complex data structure that we work with. We have a real challenge compared to some other industries. But there is potential for some types of data to be transferred in a standard format between different types of systems. Vendors and companies could work together to create an industry-wide effort.

Ms. Smith: Our next speaker is Steve Rydzewski. Steve is vice president of software development at PolySystems in Chicago. He has 30 years of experience developing a wide range of insurance software for use on both mainframes and PCs. He's also a member of the Object Management Group's (OMG) Financial Domain Task Force.

Mr. Steve Rydzewski: Making insurance systems speak to each other is a problem that every company faces. We face it on a personal level. If you've ever sent in a change-of-address form to your insurance company, the premium notice for your homeowner's policy goes to your new address while the automobile policy notice continues stubbornly to go to your old address. Even something as simple as a change of address becomes difficult if the systems can't communicate.

I'd like to discuss both the nature of the problem and two of the efforts currently underway to solve it. One effort is by the OMG, which is an international software standards organization. The other is by IBM with its insurance application architecture. In the course of this discussion I will try to answer three broad questions. First, how important is this problem? Second, current systems can do so much. Why can't they communicate easily right now? And, third, what will it take to get our systems talking?

I'm going to cover a great deal of detailed factual material in this presentation, but behind it all are two very simple points. First, there's no silver bullet to make your software systems speak to each other. There's no new magical technique. And, second, work is now underway by a number of organizations to try to remedy this problem and enable your systems to communicate.

Let's begin with just a little bit about the nature of the problem. What exactly is the problem here? The real problem is that, in spite of all the useful functionality that systems provide, our currently deployed systems can't easily work together. They're becoming roadblocks in at least two ways. First, information ends up effectively locked up in individual systems and isn't available for other uses. Today companies must commit scarce development resources just to unlock that information and make it available for you for other uses. Your own company now might have policyholder information scattered across five different administration systems, each operating on a different block of business. To pool all that policyholder information together into one form in one place so you can use it easily becomes a significant resource-consuming project. Second, because our systems don't share information readily, these systems are difficult to combine when new operational structures arise.

While I'm talking primarily about insurance, make no mistake, this is everybody's problem. It's not just an insurance problem. Every company, every school, and government agency I've talked to faces and tries to deal with this very same problem. And why should actuaries care? To be most productive, all of our companies need systems that can be modified and expanded quickly to react to new requirements. We all need systems that can communicate and work together in whatever new structures arise. This basic problem didn't just happen overnight.

Why is it so particularly acute at this point in time? It's not going to come as a surprise to anyone in this room that there's rapid changes taking place in the financial markets. More and more of our companies are becoming part of the global business community. Regulatory environments are changing here and around the world. New entrants to the insurance marketplace are increasing competitive pressure, and certainly in any ongoing consolidation of company operations, whether systems can communicate or can be made to communicate is a key issue. Your information technology departments are spending entirely too much time maintaining and integrating systems and have too little time left to provide you, the business people, with information that's of value to you.

So why is it that these systems can't speak to each other? After all, our companies have spent considerable time and capital over the years to develop these systems. The answer has at least three parts. First, historically our development focus has been on big monolithic projects. We're used to constructing entire heavyweight operational units, such as entire claim systems. Second, as that large-scale development has taken place in those companies the design emphasis has been on each isolated island of functionality. Systems were designed to accomplish specific tasks. Communicating with other unspecified systems just wasn't one of those tasks. And, third, most often the designers and developers of those systems had little or no information about how their systems might be expected to communicate in the future. In short, design and development has traditionally been done in isolation, and that's brought us to the situation we're addressing today.

What can be done about it now? What have we learned? What changes can we make given our past experiences to remedy this problem? Here are three suggestions. First, we need to change our development emphasis. Historically we've been creating line-of-business, transaction-oriented systems that are merely expected to coexist with each other. We need to be creating enterprise-wide information-oriented systems that are actively designed to fully integrate with each other. The ability to communicate must be designed in from the start. Second, in order to integrate communications into their design, the developers and designers of our systems must see the big picture. They must be able to base their analysis on the entire business domain. They must create a framework that facilitates the independent development of compatible applications. And, third, we need a consensus across company boundaries and across industries where possible on how our software components should be expected to fit together. To date there has been no consensus on hardware, software, or on operating systems and programming languages. We need to develop a consensus on the level of communications and interoperability if we're going to really fix this problem.

Two of the efforts underway that deal with these solutions are taking place with the OMG and at IBM. While the approaches they're using are somewhat different, both organizations are trying to solve the same problem. OMG is an international standards body working to develop software standards for interfaces. These standards would define how software components, both new and existing, could be expected to fit together. IBM is working to develop a new and complete framework that represents the requirements of the entire insurance enterprise. All software components developed within that framework would consistently fit together.

Let's begin with OMG. OMG is the world's largest software development consortium with more than 850 members worldwide. It is truly a global and not a U.S. organization. Members represent a full range of interests and industries including end-user companies, software vendors and integrators, universities, government agencies, and other standards organizations. OMG holds five meetings each year at locations around the world. OMG member companies are committed to the process, sending an average of about 450 people to each working meeting. So, what is OMG's approach to software standards? Formally it is to foster interoperability and portability for application integration through cooperative creation and promulgation of standards based on commercially available software.

What does that exactly mean in English? The answer has four parts. It means that, first, OMG's effort is directed toward standardizing the interfaces between software components, and it's not towards dictating the functionality of the components themselves. For example, the standard for a client information system would describe the minimum set of information that could be passed to and from such a system, as well as the request to which that system would respond. The standard would not define how the client information system should be implemented. It would not define what its full set of features should be. Second, OMG's approach is consensus based. A great deal of time and emphasis is spent on developing agreement among all parties with an interest in the standard. Standards aren't created in isolation and are not imposed by decree. Third, OMG is developing open standards that are vendor neutral. These standards must be realistic and usable, and they must apply worldwide. Finally, the standard process must quickly lead to commercially available software.

Standards should not just be theoretical constructs. This entire standards process is not just an intellectual exercise. It's intended to provide you with real benefits such as the ability to choose software based on the functionality you need and have that software work readily with software you already have. Furthermore, the standards are intended to lead to a situation where your software development resources can be used more productively, that is, providing information to you rather than just patching systems together. The OMG standards process works. Currently more than 650 organizations are building and/or adopting products that comply with OMG standards.

How are OMG's standards developed? OMG's technology adoption process is a formal structure. It's designed to ensure the quality and the applicability of the resulting software standards, but this formal process contains a large number of steps. After talking about the steps in the process we'll look at some specific examples. The standards process is supervised by a task force composed of interested parties from OMG member companies. One example of such a task force is the Financial Domain Task Force that we'll talk about later. There are other task forces that deal with other business areas such as manufacturing, electronic commerce, utilities, etc. Each task force has two primary tools: the request for information (RFI), and the request for proposal (RFP). The standards process typically begins with the task force issuing a request for information to users, vendors, and other interested parties. The RFI is quite open-ended and asks all the recipients to outline their problems and requirements, along with any particular approaches they currently favor. Based on the RFI response and the task force's own analysis, the task force would isolate specific issues and draft a request for proposal dealing with each of those issues separately. Each RFP addresses user-specific requirements and is intended to gather explicit detailed descriptions of possible standards designed to meet those requirements.

Once the RFP is drafted by the task force, it is reviewed and approved by OMG's Architecture Board. This is an internal technical group that reviews new RFPs to keep them consistent with existing OMG standards. After that review, the RFP is passed on to the Domain Technical Committee (DTC). The DTC is composed of representatives from all OMG companies, that is, the DTC that votes on and ultimately issues the RFP. Once the RFP is issued the OMG staff works to circulate it as widely as possible. The ideal goal is to solicit a response from all parties, users, and vendors who might be interested in working and helping to establish the new standard. Vendors who wish to respond to the RFP are required to sign a letter of intent. That letter binds them to completing commercial development of software and incorporating the new standard within one year of its adoption. The letter of intent is the OMG's way to make sure that the standards will quickly be reflected in commercially available software.

While the RFP responses can come from individual vendors, they typically come from groups of vendors as well. The OMG approach really is consensus based. The task force strongly encourages all respondents to work together to reconcile their differences, consolidate their responses into a single, combined, proposed standard. The task force is responsible for reviewing the revised proposals if a consensus couldn't be reached by the submitters, and ultimately the task force can only recommend one final proposal for adoption as a standard. Once the task force has made its recommendation, OMG's Architecture Board again reviews the proposed standard and makes sure that it conforms to the structure of other standards.

When the Architecture Board gives its approval, then the proposed standard returns to the full Domain Technical Committee (DTC), which is the OMG membership for a vote on approval. And, finally, given the approval of the DTC, the proposal moves to the OMG board of directors. This board must give final approval to issue the proposal as a new standard. As you can see, it's a fairly elaborate process. Typically it takes from 12–18 months to get from the initial creation of an RFP to the adoption of the official corresponding OMG standard. That's the OMG process in detail. Now let's look at some examples of the process.

One of the early results of the OMG standards process is a technology called common object request broker architecture (CORBA). CORBA is an open standard for hardware and software applications to communicate with each other no matter what programming language they're written in or where they are located on a network. Using CORBA, mainframes, client-server systems, and desktop systems all have an established standard to pass information and requests from one to the other. If two systems know what they want to say, CORBA provides the mechanism to carry those messages back and forth between them. CORBA is now in use by hundreds of mission-critical applications and industries ranging from banking to manufacturing, retail to utilities. CORBA-based products are offered by dozens of vendors including IBM, Oracle, Netscape, Sun, and Hewlett-Packard. CORBA is a concrete example of how successful this OMG technology adoption process can be.

Let's turn now to an example of how this works a little closer to home, the financial domain, an area we all operate in. OMG's Financial Domain Task Force (FDTF) has been meeting since June 1996. The FDTF's areas of interest span the entire financial domain worldwide. This includes capital markets, investment banking, commercial banking, retail banking, as well as life, health, annuities, property and casualty insurance, and pensions—basically the full gamut. In September 1996 the FDTF issued its first RFI which was narrowly focused on insurance. Nine organizations responded to that RFI. Respondents included Aetna, ING, IBM, and the Agent-Company Operations Research and Development Committee (ACORD). One clear message emerged from the responses, and that message was, we're not just insurance companies anymore; we really are financial service companies. And, accordingly, the FDTF broadened this focus to look for areas where standards would apply to all companies, and benefit all companies, in the financial domain.

What are some of the areas where standards would apply across that rather broad universe? Based on the responses to the RFI and its own analysis, the Financial Domain Task Force has developed a working list of facilities. These are facilities that are common across many software applications and many companies. Standard-setting work has begun in one of these areas, which we'll see in a minute, and will expand into these other six areas as well. The seven targeted facilities include product management, which is a repository for product definition and business rules. In this case, products would span the range from universal life plans to checking account and futures contracts.

Agreement management is a repository of contractual information about every individual agreement between the company and other parties. Agreements in this context might be universal life policies or they could be brokerage agreements. Party management is responsible for information about all the people and organizations involved with the company and the roles they play. Work is underway on this party management facility right now, and I'll have more to say about that shortly.

Financial transaction management provides processing support for financial transactions. Portfolio management maintains information about company-level investments. Investment pool management manages information about each individual party's investments separately, for example, 401(k) accounts.

Finally, there is asset and liability instance management intended to manage information about the various ancillary assets and liabilities associated with the operation of the company, for example, damaged autos recovered as salvage by an automobile insurance company. As standards emerge for each of the seven areas, you and your company will have new flexibility to select, say, a product management facility from one vendor and reasonably expect it to work with the party facility your own company has developed using OMG standards.

In addition to issuing the insurance RFI and developing the working model we've just reviewed, the FDTF has been involved in several other activities to date including the areas of currency operations and the party management facility. The currency standard was the first standard developed by the FDTF. It was adopted in July 1998 and applies to software that supports the definition management of currencies worldwide. The currency standard emerged from a joint commission from Cyborg, IBM, OTI, and SSA. Those companies are currently developing software to implement this new standard.

The party management facility is one of the seven facilities from the working model that we discussed earlier. The party management facility maintains information about all persons and organizations that have any actual or potential dealing with the insurance company. In an insurance company context, we've typically called these client or customer information systems. The FDTF issued an RFP for this facility in June 1997. Initial submissions have been received by the task force, and a joint submission is now being developed by the submitters who include Concept 5, Cyborg, Data Access, EDS, Hitachi, and Open Engineering, with support from ACORD, Genesis and 2AB. I would expect to see a standard set for the party management facility sometime during 1999.

The FDTF has been active in the areas of accounting and risk management as well. In December 1997 the FDTF issued an RFP seeking to define the interfaces and semantics required for interoperability between general ledger systems and other accounting applications, and initial submissions have been received by the task force. Submitters include Economica, Fujitsu, Real Objects, and Stanford Software. They've now developed a single combined submission. The OMG and the International Accounting Standards Committee (IASC) are now working together to ensure that the proposal addresses all relevant international accounting standards, and once that work is done a standard for general ledger systems should emerge quickly. The IASC is an example of another standards-type body that shares membership with the OMG to further this kind of standards work. Risk management represents the latest request for information issued by the FDTF. Responses to the RFI have been received and should lead shortly to the issuance of the RFP.

So far we've covered the approach of the OMG in general and the Financial Domain Task Force specifically. Let's turn now to the approach taken by IBM in dealing with the problem of making software communicate. IBM's insurance application architecture (IAA), was developed to be a high-level business architecture model for all insurance lines and a conceptual framework that provides a cross-organizational view of processing, information, and resources. This framework is intended to be dynamic and to accommodate continual change.

Industry acceptance of IAA has been good. There are 92 clients worldwide, 26 of them in North America. So, what exactly is IAA? What does it include?

First and foremost IAA is a set of business models. These models are available for companies to license and provide a blueprint for the overall flow of information and processing within an insurance enterprise. The models include such things as a data and a function model as well as a business object model, which provides a high-level, full enterprise viewpoint, and an analysis object model, which provides actual implementation details for components such as party, product, and agreement. These models were designed for use by companies when they develop their software.

The models supply the enterprise framework we talked about earlier that facilitates the independent development of compatible software components which can communicate. In addition to the business models in IAA, IBM provides solutions based on IAA. These products include: a client information integration system, which is a generic client management system; the insurance data warehouse, which is an enterprise-wide customer data warehouse; and Ensure Agent, which is a desktop system to support agent contact management. This list is likely to expand as IBM, in conjunction with its customers, seems to be moving in the direction of populating the IAA component architecture by developing and licensing other specific software components. Finally, IBM also provides the full cycle of IAA-related consulting services to help companies implement the product. Services include education, analysis, implementation, etc.

One question that immediately arises in any technical discussion on making software communicate is where do you start? Underlying IAA is a system architecture model that has proven to be quite helpful in answering that question. The IAA system architecture model categorizes software into four, increasingly finer levels of granularity. Level A, the top level, business processes, are those processes that are key to the business of the company and are highly company specific. These include information systems as well as the accompanying manual processes. For example, an overall company administration system is a Level A process in this IAA model. Level B, business systems, include software applications at the lowest level that can be implemented and recognized as such by the business. An example might be the new business system within the overall policy administration system.

Level C, business and utility components, include components of business systems that have well-defined interfaces that can be defined in business terms and are reusable across multiple business systems. An example is the party management facility I talked about earlier. That facility would be part of a new business system, part of a claim system, and part of most other systems as well. Finally, Level D, fundamental elements, represents the decomposition of the Level C components into their underlying objects, for example, an involved party object within the party management facility.

This IAA systems architecture model was included in IBM's RFI response to the OMG's FDTF and the model has been adopted by the FDTF. So, using this model, where do you start working to make software components speak to each other? First, you start with those components that are common to, and can be used by, many business systems so that your work has the widest impact. These components are marketable in their own right. To encourage commercial development of the components by a range of companies, you start with those components that have well-defined interfaces so your work has a clean, solid base. To ensure the commercial use of your work, you start with those components that can be adopted by organizations without limiting their ability to differentiate themselves from their competitors. These are the Level C components in the IAA model, and this is where the OMG's FDTF is focusing its efforts. The seven management facilities we discussed earlier fall into Level C in this model. The FDTF believes the standards that address Level C components will prove to be the most valuable to the industry.

That brings me to the conclusion of my presentation. After reviewing this material with me I hope I've been able to reinforce the two points I made earlier. First, there really is no silver bullet for this problem; second, work is currently underway by a variety of organizations to try to remedy the problem. As the work progresses it can make your job easier by removing barriers that currently exist between systems and by making resources available that can be used to provide you with information instead of just patching systems together.

In conclusion I'd like to revisit the three questions I raised at the start. First, how important is the problem? I think it's very important, and it's very important to most everyone. As an actuary, wouldn't it be ideal if the pricing assumptions you develop using System X would seamlessly flow into your administration system from Vendor Y and to your valuation system from Vendor P? Second, why can't our current systems already speak to each other? Essentially because they were not designed to do so. That truly seems to be the case. And, finally, what will it take to get our systems talking? The problem with making software work together is very widespread, and the solution's going to involve a number of parties.

We discussed the effort of two of those parties, the OMG and IBM. OMG's approach involves creating open and generally accepted standards and then having independent companies write their software in accordance with those standards. IBM's approach involves developing a complete, but proprietary, architecture that companies can use for their own software development. There remains a large amount of work to be done to define and utilize the standards that would facilitate the kind of software communication we would all like to see, but, as we've seen, that work's highly leveraged with benefits accruing to many parties.

When do we realize the full results of these efforts? Well, the fact is that the greatest gains are still years away—five to ten years is not an unreasonable estimate. It seems like an eternity in this Internet age but, taken in perspective, it's taken us 50 years to get from the very first stored computer program to the software we're using today. In that context maybe another five or ten years to straighten out one of our most widespread problems doesn't seem that unreasonable.

Ms. Smith: That was really informative. Our next speaker is Mark Orlandi from ACORD. Mark joined ACORD Standards Department as the director of company services in May 1998. He's currently involved in design and implementation of the ACORD EDI Assist product and is in charge of the Personal Line Subcommittee. Previously he was with Westfield Insurance where he was active in the ACORD subcommittee process and had experience in both personal and commercial lines, and was responsible for agency/company interface. And just in case you're thinking that he's turned up to the wrong conference because all of his experience is in personal lines, a significant amount of his presentation will be about the OLifE standard.

I wanted to add a few words and a couple of observations of my own on the OLifE standard. I have attended some of the OLifE standards meetings, and they're working on the front office systems, the systems at the agencies, and getting all of those systems working together and using data standards. But those data standards are very relevant for the back office. What needs to be developed is the OLifE standard. For any of these standards boards what is needed is the back office vendors and the representatives from the back office of insurance companies to get involved in the development of the standards for the systems that we use.

Mr. Mark Orlandi: I've been involved with ACORD standards for the last four years, and your involvement will only contribute to getting what you want out of standards in the long run. If you do get involved later, you might find it more difficult to get what you need. The sooner you're involved, and the more you're involved, the more likely it is that you'll get what you need out of the standard. I'm going to be speaking about our OLifE standard, but I think it'll give you a little bit of background on who and what ACORD is.

We're a non-profit association serving insurance companies, insurance vendors, agents, and brokers. We were formed in 1970 because a group of insurance agents in California were having a problem with personal auto insurance and they needed someone to put together a standardized paper form to make it easier and more efficient to gather that information. What's happened in the last 27 years is we went from paper forms for personal auto in California to personal auto everywhere. Then we moved on to other lines of business such as homeowners, dwelling, fire, commercial lines, and so on, but all with a property/casualty focus. It wasn't until a few years ago that ACORD got more involved with something outside of property casualty, that being life insurance. Those paper forms developed in the 1970s are very widespread.

By the 1980s the group realized that this is a great thing, but we need a way to move that same information electronically. So in the mid-1980s we developed an electronic standard, again on the property casualty side, that's known as AL3, and by the late 1980s, early 1990s, that standard became fairly widespread and implemented. By the mid-1990s, there was a technology change, which Steve just touched on, that had to do with object technology. So, in the mid-1990s ACORD got involved in developing a new electronic standard, which was referred to as AL4, but other than people familiar with ACORD, AL4 meant nothing. The term object was becoming more widespread. So AL4 became known as the ACORD object standard which, again, has a general focus on property/casualty insurance. About the same time the ACORD object standard was coming into being, there was another group outside of ACORD developing a life standard, and that standard moved into the ACORD area about four years ago. It's known as OLifE.

The standards, whether they be on the property and casualty (P&C) or the life side, have similarities. The direction is generally set by insurance companies, although there's input from other companies. It's managed by ACORD. We address program management and technical support. There's a formal change and voting process. So if you are using the standard, and it is deficient, you have the ability to present your case and say, here's what we'd like to see changed or here's what we'd like to see added. That item is reviewed and voted on within a group, and with a positive vote that change is made to the standard. We also provide support services such as consulting and training.

Specifically, OLifE addresses desktop applications. There are five key systems that most life insurance agencies deal with: contact management, financial planning, data downloads, electronic applications, and product illustrations. Not only do most agents have at least these five systems, but depending on the companies they're representing, or the product lines, they may have multiples of each of these. One of the things that these systems have in common is that they all store client data. They usually store more than that, but we'll start with client data. So you can have the same person—name, address, phone number, and so on—with that same information in at least five or six or ten different places, and let's hope that person never moves.

What the OLifE standard does is take that standard client information that's common in all these systems and brings them together into one, so that when you are updating or reading client information in one place, behind the scenes you're actually affecting it in multiple places. The best way to explain this is to draw three circles, which overlap in the center that share a common area. These circles represent applications. The OLifE focus is on integration between common applications, which is represented by the center of those three applications or circles. Think of that as the client area. All three applications have that client information in common, and that was the part of the OLifE standard that was addressed first. The other areas near the center have a little bit of an overlap and are policy data. There may or may not be some overlap or common information from one policy to another.

The other areas outside where there's no overlap would be data unique to one application. That information could include things like unique calculations that the application performs.

The focus and the integration are not only on the applications but must also address supporting multiple users of data and log-on security. Of key importance is that the standard has been successfully implemented, and it has been the implementation that has driven the standard and its requirements.

What I've described brings the data together into the OLifE object in the center. The standard has been developed to account for extensions for unique data on the OLifE server. So the server can handle not only the common data but also anything that's unique to an application or a company.

This is how it works. Let's say I open John Smith's record. In this instance, it's a contact management application and the other application containing John Smith is also opened in the background, and we have an address change. So, in the first application I make the address change, and that address change also goes to the other application in real-time. What happened was that a notification was sent to that application of what changed. Then it's up to that application, based on the notification, to decide how to handle the change. If it's something simple, like correction to a phone number, then it would go right and make the change, but this was built with the intention that you could build in business rules. For a certain type of change, not only do you want to do the change, but you want to follow up by returning some messages to the end user. How sophisticated you want to get with those business rules is dependent upon not only the application, but also the product and what your marketing schemes are.

Beyond the desktop, the OLifE standard is intended to account for distributed computing, legacy system support, service center support, the Internet and your Intranet, and being applicable internationally. Internationally the OLifE standard has been accepted by companies in Australia, South Africa, Japan, U.K., and the U.S. It works on a variety of platforms using object technology, the object linking and embedding, and the OLifE standard. Distributed-computing support addresses working in the office including a network scenario where you have multiple users on a system. It also addresses situations where you've got someone working remotely, wherever they may be. Finally, it also addresses the person who may be operating without being connected to the network. They've collected this information, and when they get back to the office, they can update what they've changed on their computer to the main database they've connected to.

The standard also addresses legacy systems not only for EDI transactions but for legacy system data. This is done by using newer technologies, such as Microsoft's DNAFS and JAVA which is applicable to J life. DNAFS is a Microsoft acronym for distributed Internet application architecture for financial systems. J life is the parallel to OLifE in JAVA, and both of these technologies make the standard applicable for the Internet, which, from a lot of companies' marketing standpoints, is a big plus.

We do have a model that addresses parties, activities, holdings, and products, including life, annuity, disability, health, and investment. The standard is 420-pages long. It is available on our Web site. You can download it. If you go to our Web site, you would get the November 1997 version. That's not an accident. We usually don't have the most current version there. The version that is there is intentionally a little old, but it's very complete. You need to be an ACORD member to have access to the latest version, which means you pay to be a participant in the standards. We can put out a version and give it away for free so that you can see what the standard is before you decide to buy in. So again, you need not be a member to have access to see what a relatively current version looks like.

The OLifE standard has only been around since1994, and it was originally developed by a group called Solutions for Life Insurance Enterprise Consulting (SLIEC). Their vision was to have an enterprise-wide life standard. They had a great idea. The problem was the technology in1994 was not there to support what they were trying to do.

In 1995 Microsoft got involved with this group and took over the direction of the standard. They worked with this group to put out the first implementation of the standard. That was version 1. However, Microsoft is not in the business of addressing and maintaining standards. So after getting on board and helping with a successful implementation, Microsoft wanted to hand the ball off to someone else, and ACORD was a natural because they had a history with standards development specific to insurance. Since 1996, ACORD took over the standard from Microsoft and released version 2. ACORD is continuing to move toward the goal of an enterprise-wide standard.

The list of vendors involved with OLifE is pretty lengthy. In fact, when I was working on this presentation, the one that was given to me was about six months old, and when I got a current list I added probably 20 vendors to the list just from the last six months. So there are a number of vendors getting more active with the standards.

Ms. Norma Y. Christopher: Is there any correlation between OLifE and the OMG?

Ms. Smith: I think that's a very important question, and I'm a little confused myself. I guess we've heard about three different kinds of standards groups: OMG, which appears to be some kind of an open standard; the IBM standard, which appears to be proprietary; and then the ACORD, which is also an open standard.

Mr. Rydzewski: All three of these organizations—IBM, OMG, and ACORD—have tried to work through to some sort of common base on these standards. Regarding the party management facility, for example, and the standards defined there, ACORD is one of the organizations that is supporting that activity. IBM has been present at a number of the OMG meetings. So I think all three organizations are trying to work toward a common base, but they are coming from different directions. OMG probably holds the widest viewpoint or perspective because it is not just insurance but software in the widest context. IBM is coming from the viewpoint of someone who has developed a proprietary insurance architecture. So my hope would be that all three organizations would continue to work together and find some way to come out with a common standard that applies rather than having different variations.

Mr. Orlandi: I can't speak about IBM, but I can definitely say that the OMG has a direct relationship with the work ACORD has underway concerning its object standard. When I began I mentioned we have an object standard for the P&C, and within the last two-and-a-half years the OLifE standard came on board with ACORD. Internally, with OMG's involvement, we've been working to bring the object standard, which was originally developed for just the P&C side, to work with the life side so that when we speak about an object standard we're encompassing everything for insurance and, for that matter, the financial industry as well. The goal is definitely not to wind up with separate standards that don't speak with one another. OMG has endorsed the work that ACORD is doing concerning its object standard and what the OLifE standard is doing in association with the object standard.

Mr. Rydzewski: My understanding is that ACORD is a member of OMG, and OMG is actually a member of ACORD. So that coordination is taking place.

Mr. Jeff Robinson: I'd like to thank the panelists for one of the most practical sessions I've ever attended. I've spent about 35 years trying to have systems talk to each other, and the biggest frustration is actuarial data, which every system needs. This is an area in which I think the actuaries really have to be involved, because actuaries know what's needed in a life insurance system. They should have the overall view of the situation. Michelle, can you tell me what the Computer Science Section is doing in this area? This is one of the areas we should lead in.

Ms. Smith: This section is promoting the use of the table standards. So one area is the table standards. And I think we would get fairly widespread buy-in on that standard because it's such a natural way of organizing the data for a table. But as for other standards, the Computer Science Section at the moment is not involved in any of the others, like policy data, as a group. But I think we need to think about that.

I encourage everyone here to learn more about the standard and what they do, and to get involved in the development of those standards. The problem at the moment with the OLifE standard is that there's no representation from the back office. They have the standards developed for financial planning and illustrations, but an actuary may come along from the back office of a company and look at what they've already done for illustrations and say, we really want something different for our policy object. Getting involved earlier you really have a much bigger say in what's going on. I think I'll speak to the Section Council about how we might encourage people to get more involved in the development of standards. Joining ACORD, for a vendor, I think costs about $1,000. This is just to be a member and to participate in the meetings that help design the standards. If you're a company, I think it costs a little more.

Ms. Krueger: To stress what Michelle just said about company participation, vendor participation has increased greatly. I'd like to see more increase on the company side, and that is who is going to drive the resolution of the standards. It's not going to be the vendors. The vendors will participate, but the companies are really going to need to participate to get what they want.

From the Floor: Does OMG and IAA have web sites where you could pick up the RFIs or RFPs to see what they're talking about?

Mr. Rydzewski: Certainly. They both do. I don't know the IAA address, but the OMG address is www.omg.org, and that will lead you to all sorts of material about the OMG. I know the IBM Web site. If you use a search engine, you should be able to turn up something for insurance application architecture. They have a fair amount of information available on the web as well.

From the Floor: I have a basic question for Mark about the actual data itself. Are there still separate databases for each application, and will OLifE funnel that information to each of the applications? Or is there a common database that they all access to get the same data, for instance, the John Smith address change?

Mr. Orlandi: My understanding is you still have the separate systems in the background, but from the user's perspective they're one.

Mr. Robinson: I think one area that the actuaries could help in is product feature standardization. That's something we're all quite heavily involved. Every systems needs a listing of the product features.

Ms. Smith: ACORD is already working on the standards for illustration systems. That's what I'm saying. If actuaries were involved now, we could have more say in that development.

From the Floor: What do the standards organizations do to monitor or control commercial claims of compliance with their standards?

Mr. Orlandi: We have a certification program or process, and this is not just true of OLifE, but for all of the ACORD standards. If you are developing something that you believe is in compliance with a standard and want to promote it as such, we publish a listing of who has been certified by ACORD.

From the Floor: Software seems to be a key issue now. As we move from a domestic-focused environment to a more global environment, I'm wondering what the standards organizations are doing with regard to identification of individuals? For residents of the U.S., of course, where it's available, the social security number is clearly a definitive identifier, but what do we do in a global environment to determine whether this Jim Smith is identical to the 100 other Jim Smiths who may be applying?

Mr. Rydzewski: That's a good question and not a simple one. Within the OMG context what happens is that rather than defining a social security number as the identifier, there's an identifying number. It might be a social security number, it might be a social insurance number. It could be whatever other number you might have depending on the particular area you're working in. The broader question of identifying which John Smith is not a simple one.

Mr. Orlandi: I would agree with Steve. Depending on whom you are communicating with, that may or may not be an issue because regardless of who you are sharing data with may have come up with their own method of addressing it. Preferably the standard would address it. My understanding is that with the OMG's involvement that is one of the issues they are addressing. The OMG is literally global in its management of this object technology and addressing things to make them that interoperable.

Ms. Smith: This question came up at a recent OLifE meeting that I was at in June 1998. Every country has some kind of identification number. So, an identification number is built into the standard.