

RECORD, Volume 25, No. 1*

Atlanta Spring Meeting
May 24–25, 1999

Session 3PD

Data and Interoperability Standards for Actuaries

Track: Computer Science
Key Words: Computer Software, Research, Standards of Practice

Moderator: MICHELLE D. SMITH
Panelists: JEFFREY G. ALLEN
MICHELLE D. SMITH
STEPHEN J. STROMMEN
Recorder: JEFFREY G. ALLEN

Summary: Open data standards are developed for the purpose of enabling easy communication and transmission of data between different systems developed by different vendors. The panel discusses open data standards: how they are developed, who develops them, how vendors' systems become standards compliant, and how standards are normally "spread" across an industry.

Panelists also discuss the development and current status of data standards in the life insurance industry affecting systems used for financial planning, illustrations, administration, and valuation. Particular attention is given to the table data standards being promoted by the Computer Science Section of the SOA. This includes a description of the data standards, vendor participation, and projected future of the data standard.

Ms. Michelle D. Smith: It would be interesting to know how many of your companies are working with any kind of data standards in your systems. An example is Agent-Company Operations Research and Development (ACORD), a committee of Independent Insurance Agents of America. ACORD is a data standard that's fairly widely used in the front office by agency systems.

I think that what we're trying to do with data standards within the Computer Science Section might overlap with what other data standards organizations are doing.

I'm speaking because the Computer Science Section has spent the past six to eight months promoting data standards for mortality table type data. Steve Strommen, who will speak after me, was on the Section Council in 1994-95 and developed standards for mortality table data. Those are the data standards that we're trying to promote to vendors and to insurance system users.

* Copyright © 2000, Society of Actuaries

The handout that you received is a copy of the white paper developed by the Section regarding data standards. The paper describes the table data standard that was developed by the Computer Science Section. At the back of the report, there's also a survey to solicit input from software vendors and insurance system users on how these standards should be developed in the future. The survey will help measure if there is sufficient interest in the data standard and whether vendors and other users want the standard to be adopted. Feedback will be solicited regarding how the data standard might evolve over time, what kinds of additions might be made, and what steps should be taken to encourage the adoption and further development of the standards.

I will give an overview of what data standards are and why we support their use. Then Steve will talk specifically about the table data standard that's described in the white paper. He will give a demonstration of TableManager. A version of TableManager is available on the SOA's Web site. TableManager has many mortality tables stored in the standard format that we are discussing. I will give a demonstration of a projection system that is already compliant with the data standard so that you get some idea of what standards compliance will actually mean in practice when the system is adopted.

The primary reason for data standards is to enable data to flow smoothly between the different computer systems of a company. To understand data flow, you need to look at the different data flow processes in the company.

Of course, companies have multiple information systems. For example, there are product development systems, pricing systems, and sales systems. Much of the information that's processed in product development also goes into other systems, such as the administration and illustration. For the actuary, there are valuation, projection, and experience study systems. Currently, there is a lot of common data flowing between all these different systems, but the flow of that data is not very smooth.

If you just think about mortality table data, the pricing process is going to produce many rate tables. That data needs to flow into all these other systems. Generally, the table data needs to flow into the illustration, sales, and administration systems. That same table data needs to flow into the reserve, valuation, projection, and experience analysis systems.

Let's look at what we want to accomplish in a fairly rough overview of how data flows through the various systems, from one system as output into another system as input. One of the outputs of pricing will be a lot of table-type data, like premium rate scales or cost of insurance (COI) scales. That information needs to flow into illustration systems as input and into valuation and projection systems. The current situation is that each one of these flows going from output from one system to another is often done differently on a case-by-base basis.

It may be done by re-keying data, that is, taking the output data from one system and keying it into another system. Data are coming out of one system in one

format as output, and it needs to be input in a different format into another system. This is what creates major inefficiencies in data flow in the insurance industry.

There are many reasons why that process is so inefficient. Each data flow represents a different process, and it can be very labor-intensive work to get data from one system into another. Of course, one of the biggest problems is that companies have such a variety of systems doing different things. Different vendors might have designed the different information systems in a company. They might have been created internally by different people at different times during the life of the organization. There might have been an administration system created by one vendor who specializes in administration systems. Another system may be a projection system created by a different vendor with that specialty. Each vendor may tend to have a narrow focus and not have the perspective to coordinate how data might flow more smoothly between the systems. This kind of coordination needs to be encouraged for data systems applications in the insurance industry.

There have been patches developed for each of the data flows between processes. Custom patches are often developed for the output to be input in each situation. Efficiency would be enhanced if standard formats were used to make the transfer of data go more smoothly. For example, once output like product specifications, premium rates, or COI scales have been developed, they should be able to flow smoothly into all the other systems.

There are various ways to do this. One of them would be to have one massive system that does everything. I'm not sure if that's even possible. The other way is to create an interface for each of the processes. For each flow, there would be an interface that would connect the data. Each company would have to do that individually. The best way to do this is probably through creating a data-sharing structure. If the industry is able to assist in the development of such a standard, which would help this interface, efforts will not be duplicated, and the cost can be spread across the industry. Instead of having company-wide data standards or interfaces, industry-wide data standards can better spread the development cost across the industry.

The big system idea isn't really practical. You're either going to have total reliance on one vendor or you're going to have to do it in-house. That will be very expensive, and will not be very flexible. It will require the creation of custom interfaces between each system.

There's a possibility that you will still need to hire outside vendors to help develop custom interfaces. The best solution for the industry as a whole and for each individual company would be to use data standards. This means developing a standard data format for each type of data, so that data will flow smoothly among systems irrespective of the vendor, the operating system, or the platform that is being used. But that really requires a lot of cooperation, coordination, effort, and dedication from everyone in the industry.

The best way for data standards to be developed is for everyone in the industry to participate and to have them developed democratically. This is different from what

we often see in the development of other data standards, where large companies, to their advantage, tend to dictate how standards should develop. The best thing for an industry is to have many different vendors and users involved in providing input to the data standards. When a data standard is developed democratically, the standard will be more open, meaning that it becomes public information. Any vendor has access. The standard can be incorporated into an organization's software without having to pay to use it. Another issue regarding standards that has been apparent in other industries and areas of the financial services industry is that they are increasingly international. Since these organizations and vendors are becoming more international, they have to do a significant amount of cross-border transactions. They might as well be international standards.

ACORD is a non-profit organization that has been around for over 25 years. They have developed data standards. They started working on forms, particularly for the property and casualty industry, and then they moved over to the life industry. They took over responsibility for the life industry, which used to be handled by Microsoft. Microsoft wasn't interested in it anymore because it was specific to the life insurance industry. Presumably, Microsoft wanted to focus more on the other financial services industries. ACORD took it over. The standard that is being developed by ACORD for the life insurance industry is called OlifE. That standard is currently being implemented in illustration systems and financial planning software.

There are a lot of data variables that are used in illustration systems that overlap with the data we use in product development, pricing, and valuation systems. ACORD has many years of experience in developing data standards. I'd like to give you some idea of the process that ACORD uses in developing data standards.

As part of the democratic process that is involved in developing these standards, the first thing that happens is some sort of association (a non-profit group or a consortium of participants who are vendors and systems users) is created. Every time a data standard is needed, a task force is established to head up the creation of that standard. For the standard to evolve over time and to incorporate new things or make improvements, members of the association submit suggested changes. Once these are submitted to the task force, members will vote on suggested changes.

If the majority vote "yes," then the change will be incorporated into the standard. Technical experts will analyze the change to make sure that the change is technically correct and in line with the goals of what the standard is trying to do. The standard can then be adopted or the change can be adopted into the standard, and system vendors can implement the standard.

It takes a lot of effort, and to be successful, the industry, as a whole, must really want this. The Computer Science Section has been trying to promote this for some time. Based on the attendance at this session, I'm not sure that we have the commitment within the Society to really want this to go ahead. The Computer Science Section's plans are to distribute the white paper to vendors and get their input. Some vendors have already started to adopt the standard.

We also obviously need an association developed to oversee the standard. It will require rules and regulations as to how changes to standards get submitted and how they get approved. The biggest criticism that we have received is that the standards are not as useful as they could be because only certain types of tables are included. For example, the current standard does not include disability tables.

The standards must be considered a work in progress. The white paper that we've put out is Version 1.0. It's our first version of the standard. When a version is final, vendors can start adopting the standard, but that doesn't mean that the standard won't evolve over time. At the moment, we have regular select-and-ultimate tables included, but as we get additional input from users and from vendors, we'll start adding more tables. It's always going to be a work in progress. If you try to develop the standard and include everything before it is released, it will never get completed.

Once a version of the standard is ready to be incorporated, vendors can adopt it voluntarily. There will be an incentive to adopt the standard if big vendors adopt the standard early on. Hopefully, momentum will gradually increase, and other vendors will adopt the standard based on the benefits and efficiencies of being compliant with the standard. Once vendors have adopted the standard, it is important to have some method to independently verify that a vendor is standards compliant. Many associations, like ACORD, provide for independent verification or certification that a system is compliant with their standard. The association would have to come up with a definition of what compliance means and how one can test for it. Once the vendor is certified as compliant they should be able to advertise this in their sales and marketing.

Steve Strommen will talk about the table data standard that has been developed by the Computer Science Section. Steve will demonstrate the TableManager program, and then I will demonstrate how one system is becoming compliant with standards.

We want to talk about how we might go about developing the standard. We would like to get input from people who are interested in using a standard and also from vendors who are interested in making their systems standards compliant. One of the things we'd like to discuss with reference to the table data standard in particular is, how would we develop some kind of an association group that could develop the standard. Another is, how would the standards be monitored for changes going forward? What is the commitment to getting this done? Also, what do people think about compliance verification? Do we just use the honor system, which doesn't cost any money, and is not very reliable, or do we come up with a way of independent verification?

Another issue that needs to be raised is that the TableManager includes SOA tables, and tables from population statistics from all around the world, but the data hasn't been fully checked. Ideally, we would like to have a higher level of verification as to the accuracy of these tables. At the moment, no one is prepared to do that. We might want to talk about how we could get that done as well.

There are other issues such as platforms for the TableManager program that Steve's going to demonstrate. What developments beyond table data should be considered? There are many additional kinds of data that don't come under the table-data type. For example, policy product features might be considered. Standards could be developed along with ACORD for back office systems.

Mr. Stephen J. Strommen: I'd like to bring all of this down to a hands-on level. The TableManager had its genesis with an assignment that I got back in early 1994 to evaluate certain tax reserves. In order to calculate a bunch of tax reserve factors I found myself developing a spreadsheet of the 1980 CSO table. I thought about how this is kind of dumb. How many thousands of actuaries have had to do this exact same thing? Shouldn't there be a way to get table data into spreadsheets or into any other software so that it's easy to use, and so that you know you have the right numbers, and you didn't mistype something?

With that genesis, and as a member of the Computer Science Section at the time, I started an effort towards a mortality tables database. I'll be talking about some of the history behind the development of that TableManager. Then we'll do a demonstration. I'll get into technical details which I hope don't completely bore you, but they are important because when you develop a standard, it tends to be very detailed and specific, and you'll see some of the kinds of issues that come up in that kind of effort. We'll talk about the formats, programming interface, and then list the issues to be addressed. We will also have some time for discussion.

As I mentioned, the TableManager had its inception when I had this little tax reserve assignment problem. I had the idea of developing a database of mortality tables. That idea was expressed in an article that I wrote in 1994 for *Compact*, the newsletter of the Computer Science Section. There were a number of letters to the editor and other responses that expanded on what ought to be included in such an effort. A second article was written, and we eventually decided that the purpose was to develop a standard way to share table data in electronic form.

In doing so, we employed an object-oriented design and implemented the table as an object, which is something programmers can use in a large number of different systems. The proposal was refined and implemented in early 1995. That's when the first version of the TableManager was put out on Actuaries On Line. We didn't have the Internet site at the time. Version 1.0 turned out to have a couple of bugs in it. Version 1.1 followed shortly thereafter and has proven fairly stable and usable since then. In the years since then, the idea that the database ought to be developed through voluntary contributions really took hold with a few people. In particular, Roger Scott Lumsden has contributed hundreds of tables to this database. He always includes the caveat that the numbers have not necessarily been independently verified by an outside authority. Very few errors have been found in them. Roger has contributed hundreds of tables. Robert J. Johansen has contributed all the versions of the 1980 CSO tables. The male/female, all the blended sex versions, age last birthday, age nearest birthday, smoker/nonsmoker, and all the permutations thereof were contributed by Robert J. Johansen. He has also contributed a few other tables.

We now have, I think, four different databases: one for annuity tables and projection scales, one for life insurance tables, one for population tables, and a whole separate database just for CSO tables. Volunteers contributed all of these tables. The software and these databases have been downloaded many hundreds of times and are in reasonably wide use. I can't say the majority of actuaries use it, but in our small sample, at least, we had one person out of less than ten that has used this software.

What I'd like to do at this point is give you a little demonstration of how Version 1.1 looks and works. The program is a Windows stand-alone program that operates on a database of tables. When you bring it up, the very top line tells you what database you're viewing. This one is called q_x . That database has a large number of tables in it. If I go to the table menu item, and I say I want to view a table, we get a list, which goes on for a very long time. It starts with the 18th Century Northampton Table. If you want to pick out the 1975-80 U.S. SOA's Basic Tables, you will have the male and female, age last birthday, and age nearest birthday. You can highlight the one you want to look at, press the O.K. button, and get an image of the table. This is essentially the table data standard format that we're talking about developing.

We're talking about a text file that has certain tags on the beginning of lines. The table name with a colon after it is a tag. After that you've got your table name, table number, table type, who contributed it, and the construction method. There are a lot of tags. Some are required; some are optional. For example, the construction method is optional. There are other optional items like the observation period and the unit of observation. Those are not all included in every table that gets contributed this way, but there are at least provisions for that kind of thing in the standard data to format.

You'll notice that the documentation under construction method takes several lines until you get to the next key, which is the published reference. The provision of the data standard is another detail. You can use several lines for any item. You can put in comments. And then there are required items like the minimum age, the maximum age, the select period, if it's a select table, number of decimal places, and then the table values that are laid out as they typically are. The first duration, if it's a select table, is called "Duration 1." Some people like to call it "Duration 0." The program then allows you to extract a table to an external file. You may want to do that. You can take it out of the database and put it in a text file for use by another program that might be able to use these tables, but, for whatever reason, you might not be able to use the database directly. If you did that, you'd choose the table you want. We'll just do this with the Northampton Table. You have to tell it the name of the file you want to create, and because the Northampton Table is Table 250, it gives a default name of T250.TXT, and if you want to create that table, you can say O.K. It tells you a table has been created.

This program then allows you to open a text file. You can take a table that has been created this way, and use it as a template if you want to create a new table. In this case, we'll just create Table 1 because I know there is no Table 1 in this database. If we save this, we can now import it into the database. You could use

this as a template and then type in whatever numbers you had in your own table. If we wanted to insert a table into the database, we saved it as T250.TXT. It will say the table is successfully added to the database. If you go in and you want to view a table, I believe it's going to wind up at the end because this is sorted alphabetically, and letters come after numbers. You can easily add tables to the database and extract them and use them. You can extract them in a worksheet form as well. That's basically the idea. You can have an entire database of tables and easily extract them or put tables into the database.

The text format of these tables is really the data standard that the Computer Science Section is advocating, and that is being widely used at this time. It's very simple. You just have a tag at the beginning of each line. If a line doesn't have a tag at the beginning, it's just a continuation of what came before. Those tags have to end with a colon, and there must be specific words before the colon. If you want Table Values, it has to say Table Values and then a colon. You then tabulate your values. They can either be a select or an aggregate table. If it's aggregate, you don't have a line that shows the duration.

There is a spreadsheet format that is almost the same thing, except obviously it has to be set up in row and column cells. In this case the tags go in column 1. The descriptive text or other data goes in column 2. If there is more than one line of descriptive data, you can have more than one line in column 2. The table values get lined up at the very bottom in spreadsheet form on a spreadsheet.

There's also a binary format that is what is used inside the database. In the binary format every table is a series of what I call generic variable-length records. Every record has three fields. The three fields are a numerical code for the type of data, the length of the data in bytes, and then whatever data it is. This kind of record format is widely used by a lot of software. For example, Excel and Lotus spreadsheet files are both stored in this kind of a record format. You can put any kind of data in a file that has this kind of a record format. In the case of a table, spreadsheets, and other data that uses this kind of format, the last record has a special numerical code that indicates this is the last record. You have reached the end of something.

The database format is basically a set of these tables put end to end. They're stored in the file one right after the other. There is a separate index file that tells where each table begins and how many bytes from the beginning of the database you have to go to get to the start of each table. The index file has three fixed-length records for each table. It has the table name, the table number, and this offset from the beginning of the database file where the table begins.

The reason I'm going into some of these other things beyond the text file format is that these are optional other things that can be involved in a data standard. You can have a binary format for the same data so you can have a database. You can also get into a programming interface.

How is a system going to access these data? What was done when the original TableManager that was put together was not only defining the data but also making

available programming interfaces. The letters API stand for application programming interface. There is a C++ application programming interface for a table that basically defines a table as an object with certain methods and properties that you can use to act upon it. The most important ones that would be used in a system if you were writing something would be the ones that get a value based on an age or an age and duration.

The ones that read and write a table are also important, either in text, spreadsheet, or binary file format. There are other methods and properties that are involved. The code for this programming interface was made available on Actuaries On Line and is currently available on the Web site.

There is also code for the table file that represents the database. We were talking about the table, which is a unit or one item in the database. The table file class has a set of properties and methods that include the ability to set the file name; it either opens the database if it exists or creates one if it doesn't. It will append the table to the end of the database, get a table by number, delete a table, or compress the file, which is necessary if you delete a file from the middle of the database.

There are miscellaneous other things. These are the kind of operations that an actuarial software developer would need to be able to access. There was also an a programming language (APL) programming interface that was put together, and in that interface, a table is represented as a nested array. The APL language is not object-oriented, so there needed to be another method of addressing that issue. In the nested array there's one element of the array for each data item, and one element, the last element, holds the values in the table as a vector. There are functions in APL that are provided to get and set values in the table and to obtain the values from a database file. The text-file in the spreadsheet formats is not supported in the APL format. The idea was basically for the APL programming interface. The idea behind this was just to provide APL users a way of getting values out of this database.

There are a number of questions that need to be addressed when we get to brass tacks like this. For example, there are limitations in the current format. It requires values to be in the table at every age and duration. You can't have a quinquennial table. The software won't handle it that way. There is only one set of values in each table. For example, if you wanted both q_x and l_x in the same table, this format doesn't allow for that. You've got to pick. It doesn't allow for parametrically defined tables. There are other kinds of tables, like disability continuance tables, where you typically have values for 24 months and then annual values thereafter. It doesn't provide for that. There is the question of file formats. Should the text file format be the only one that is the standard or should maybe the binary format or the database format or the spreadsheet layout become a part of it?

There are implementation details. Should this application programming interface that I talked about be available in other languages like PASCAL or JAVA? What platform should it be available on? If you're implementing code, it becomes important if it's 16 bit or 32 bit. The APL approach can be addressed in different

ways. You can provide a source code that has been done so far or you can provide something along the lines of a server or a COM-object, which is a precompiled module that can be incorporated into software by other developers.

There is one issue in the binary format that has been used so far. A hash value was used for each table as a means of verifying that when the computer got the information off of a disk, it retrieved the right numbers. That was felt to be more important back in 1994-95 when data checking, Internet transmission, and on-line transmission were perhaps not as safe as they are now. There's sort of a check value included with each table. At this point in time it's a marginal value.

Michelle touched on database maintenance details. Should we have one database or many? What is more important is who ensures the accuracy of these numbers? As I mentioned, many of them have been contributed by volunteers. They haven't gone through a third-party verification process. Many of the tables included a published reference in their documentation that you can use to go and check whether they're accurate, but that sort of defeats the purpose, and so there ought to be some kind of assurance of accuracy. Before we get to the discussion, Michelle would like to demonstrate one application that has already been made standards compliant.

Ms. Smith: The white paper talks about two levels of standards compliance, and the one that we're really pushing in the Computer Science Section is just the bare bones level of compliance, which is compliance with the ASCII format of the table data standard. The more comprehensive level of compliance would be to actually have a system that can access the tables directly in TableManager. What we've done in TAS is to develop compliance with just the ASCII format. For this demonstration, I'm going to load a different database. I'm going to load the annuity tables database, and then extract a table. I'll get one of the 1983 individual annuitant mortality tables. There'll be a separate icon on TAS where you can import and export the tables in a standard ASCII format. A little bit further down the track we intend to make TAS able to pick up tables directly from TableManager using the C++ API that has been provided with TableManager. At the moment we have everything working with the standard ASCII format.

What I could do now is take that table that I just exported from TableManager. TAS has a whole bunch of table types: termination tables (which is the one we're using at the moment), premium tables, expense tables, and so on. There are 20-50 different types of tables.

This describes what came in from the TableManagers, and then we have all this other information that you saw in Steve's demo with the standard ASCII format that comes through here as well. If I wanted to export a table from TAS and go in the other direction, I can enter all this information and then put that table into TableManager. When I save this file, I can actually save it as a TAS table. If I open one of my product data files in TAS (and this is a termination table), I'm going to put it in the withdrawal section, and that now is in the proper TAS format.

I can also work in the other direction, and if I wanted to, I can export a table. I've set up a table here, and I'll just show you what it looks like in TAS. Then I could export this into the standard format.

This is a bare bones level of standards compliance that we're talking about in the white paper that we're encouraging vendors to adopt. Obviously, it's going to be much nicer when we have it set up so that TAS can directly go and pick up a table right from TableManager. That's what will ultimately happen. We're encouraging vendors to do that. But first and foremost, we'd like to see the standard being incorporated at the bare bones level.

The white paper developed by the Computer Science Section has been reviewed by several groups within the Society. Because this standard is being developed within the Society, and we're sending this white paper out to vendors with the Society's name on it as part of the Computer Science Section, certain groups have wanted to look at the paper and see what they thought about it. Jeff Allen can provide an update on who has looked at the white paper and the comments that have been made.

Mr. Jeffrey G. Allen: Michelle and the Computer Science Section have made it a priority to communicate with other groups within the SOA and to coordinate with them, and get their feedback. I'm going to run through some of the groups that have looked at the white paper. Early on, the SOA Research Committee, the SOA research staff, Dave Becker, SOA Vice President of Research, and Sam Gutterman, Past-President of the SOA, were involved in reviewing the paper.

Sam provided some recent feedback that we're in the process of reviewing. Legal counsel also reviewed the paper early in the process. The disclaimer included on the sixth page was added as a result of this review. This issue was related to the Computer Science Section providing a list of "compliant" vendors or companies. The disclaimer clarifies that the SOA does not endorse a given vendor or company.

The Technology Committee, chaired by Steve Strommen, has been closely involved. Historically, their focus has been mostly on internal technology issues within the SOA. This paper deals with more of an external issue; however, the committee was willing to provide a review. The paper was distributed to the Operations Committee of the SOA.

Finally, some individuals have been requested to review the paper from the perspective of the American Academy of Actuaries and the Actuarial Standards Board. One of the issues identified was the sensitivity to the term "standards" within the actuarial profession. There is a desire that we clearly define what it is that we're talking about when we're talking about "data standards," and that they are distinct from Actuarial Standards of Practice. Those most involved with data standards see the clear distinction between the data standards and actuarial standards of practice. However, at a more surface level we need to make sure that communication regarding these standards is clearly defined to avoid any possible confusion.

Mr. Alan G. Dickey: Without speaking for my firm, PolySystems, because I really don't know what the official posture is, we have no reluctance to the standards. I don't think we've been active in establishing the standards, but I see no opposition from our firm as far as adopting them over time.

Ms. Norma Patz Fox: As a reinsurer, we are typically forced to take data from a lot of different sources. We're all for standardization, and I think I would encourage our vendors to go that way, and not only with mortality tables. I think I'd spread it out even further.

Ms. Smith: A couple things that we've been grappling with is what sort of organization should we do this in? Should the computer Science Section take this on as a project or should it be done inside the Society or outside of the Society? I don't know if anyone has any feelings on whether the Society should be involved at all.

Mr. Strommen: I'd like to see a show of hands. How many are members of the Computer Science Section? The Computer Science Section does have a treasury, in which there is some money that could be spent on this kind of a thing. One question is, should some of those funds be directed toward this project?

From the Floor: I guess my question is if the actuaries don't do it, who's going to?

Ms. Smith: There's one reason I've been trying to push it within the Society. When I look at groups like ACORD, which are already working on data standards for illustration systems and financial planning software and so forth, I see that they're already starting to think about tables. If we don't do something, as actuaries, we're going to end up in a position where it's dictated to us exactly what the standard will be. It's not just for table data; it's for everything. If we're not active now and we don't get some commitment from the other standards boards, it could cause a problem. I think it will happen eventually, but I think it will happen in a way that we don't necessarily approve of because we don't provide any input.

Our next plan is to do a mailing to vendors. At this point we pick the major valuation, administration, projection, and illustration system vendors, and send the white paper to them. We'll ask them to fill out the survey and give us input on how they think the standards should be developed and whether they're open to adopting the standard. Once we get feedback from those vendors I think we'll have a better idea of how to proceed in the future.

One thing that we have done is send the white paper to ACORD, which is working with these other data standards for the front office of the agency systems. They've said that ultimately when they get up to table data, they will be prepared to adopt the standard, and that's what needs to happen. We don't need two table data standards in the industry. It would defeat the purpose of having standards. That's promising as long as we keep reminding ACORD that we have this table standard. The only way we can do that is if we get enough support within the actuarial profession to keep this thing going and to get vendors to start adopting it.

I don't know if anyone has any feelings on other issues like how it should be organized, how the task forces would be structured, and how submissions would be taken for this.

In terms of funding, if it was decided that the Computer Science Section would be the main body that deals with this, then that could become a primary function of the Computer Science Section.

Does anyone have any thoughts on compliance? The cheapest way to do compliance is using an honor system. The vendors would declare whether they are in compliance, and they would be listed with other compliant vendors. This would involve no independent verification that the vendors are actually compliant with the standard. The only advantage of doing it that way is it doesn't cost anything. If you come up with an independent verification system, then you need to actively test the system. You need to have software tools to test the system, and a procedure to do that. That's going to cost money, so you need funding for that as well. You need funding for the standards development, but also for the verification. I think that would mean funding from the members, vendors, and users.

I want to give you some idea of the magnitude. For ACORD, I think the vendors pay \$1,000 for membership each year. That's the system vendors. They're the ones that have to supply the resources and time to do the additional program to make the systems standards compliant. The users should be charged something like \$10,000 a year to be members of ACORD. This allows them the means they provide input on the development of the standard. Of course, it is the users that want this thing the most. In the scheme of things it's not a huge amount of money. That's for a much broader standard that covers much more than our table standard.

If you're talking about funding for just the table standard, and the maintenance of the table standard, I would think that it's going to be less than \$1,000 for each member company. In the scheme of things, it's not such a big deal, but we do need companies to buy in, if they want independent verification of compliance.

It's important to try and get some momentum. I think once people realize what this really means, they will be excited about it. The most common criticism is, like we said before, why don't you have disability tables? This is still in the early stages, and other tables might be included in Version 3 of the standard. It's always going to be a work in progress.

Mr. Strommen: I should say the Technology Committee's initial reaction was very favorable. Nobody found anything of concern, except, as highlighted by Jeff, the reference to the word *standards*. There was a little touchiness about the use of the word *standards*. Maybe we could call them *formats* or something. The Technology Committee was very much in favor of it.

Ms. Smith: I guess that the other issue related to the terms of the honor system for compliance verification is, what are we going to do with this database of tables?

Tables are submitted on sort of an honor system. If someone submits them, you just assume that the table is right. For people to fully use the TableManager will require some sign off to indicate that the tables are fine. That might be through the SOA for the SOA tables and the Institute of Actuaries of Australia for all the Australian tables. That requires approaching each organization, each society, and probably funding for that. You need a person that's going to be able to sit there and check all these things and make sure they're right. Another thing we need is some commitment. I think if we came up with the funding to have that done, it would be important. It's something that everybody wants, but no one wants to do it.

TableManager only runs in Windows right now.

Mr. Strommen: That's the only platform on which it has been implemented. Any system uses almost exactly the same text file format, whether it's an Apple, Unix system, or Windows. The file format that we're talking about is very broad-based, but, as yet, there is no software or even a programming interface written for any other platform other than Windows.

Ms. Smith: I think that the platform is another area where modifications might be possible if supported by a data association.

There are other types of tables that people might want included. It's important to develop the standards, so that, at some point, parallel standards aren't developed by some other organization. We're going to have to collaborate with all the standards bodies and make sure that we can use whatever they've developed so far in terms of products and standards for product features. For example, ACORD has developed all these standard definitions of objects for life products with all the product features and so on. It hasn't got up to table data yet. We could at some point think about vendors using that standard for back-office systems and then adopt the table data standard.

Mr. Strommen: I have another comment on the idea of momentum. There are some other things going on currently with regard to the TableManager. Last Friday a new version of it, which is an Excel add-in, was put out on the SOA Web site. Now there's an Excel add-in, which gives you Excel functions, which will provide access to any value in any table in the database via an Excel function. Second, there's another session at this meeting (2OPD) called The ABCs of component object model (COM) that Microsoft uses. At that session, which will not be included in the *Record*, one of the presenters will be showing or demonstrating a COM object that represents a mortality table using this format. There are some other things that are going on, and there's some momentum behind the standard already.

Ms. Smith: Does anyone have any suggestions for vendors that we should definitely pursue? Somehow we need to raise more public awareness within the Society of what this is really all about. It's something that's pretty useful. I think the white paper is actually going to go on the Web site.

**Proposed Data Standards for Table Data
in the
Insurance and Pension Industries**

*A white paper describing a standard table format for use in
information systems that use or produce table data and the
advantages of adopting the standard.*

Version 1.0

The Computer Science Section of the Society of Actuaries
January 1999

Table of Contents

| | |
|--|---------|
| Overview | Page 17 |
| Background | Page 18 |
| Why use TableManager and the SOA ASCII Table Format? | Page 18 |
| Who Should use the SOA ASCII Table Format and TableManager | Page 19 |
| How To use the SOA ASCII Table Format and TableManager | Page 19 |
| Current Known Limitations | Page 20 |
| Future of the Standard | Page 20 |
| Company and Vendor Support | Page 21 |
| Glossary | Page 22 |
| Further Information | Page 22 |
| Appendix A -Description of the SOA ASCII Table Standard | Page 23 |
| Appendix B - List of Other Insurance Industry Data Standards Initiatives | Page 25 |
| Appendix C - Survey on Implementation Intentions and Future Development Procedures | Page 27 |

Overview

Data standards are developed for the purpose of significantly improving workflow by facilitating the transfer of data between different systems from different vendors. This gives companies the flexibility to choose "best of breed" software packages for different purposes and still have them communicate to one another. Such standards allow developers to build software applications to a common data model, thereby allowing users to focus more time on actually using the functionality of the system, the results produced, and less time on manipulation of data formats in moving data from one system to another (or, as is sometimes the case, in re-keying the data).

Data standards are becoming prevalent throughout many industries, including financial services and the insurance industry in particular. (See Appendix B for a list of Data Standards efforts within or affecting the insurance industry.) The actuarial community stands to benefit greatly from participation in the development of these standards, and ultimately in the implementation of such standards.

To date, the actuarial profession has not played an active role in any of the major standards development initiatives. Much attention has been focused on systems used by the agency force, including data used by illustration systems and data stored in administration systems. These initiatives are then defining data standards for all kinds of policyholder data and product feature information. This is the same data used by actuaries and other financial professionals in the back office for the purposes of valuation, financial reporting, projections, experience studies, product development and so forth. It therefore seems logical that the insurance and pension industry data standards already being developed are ultimately extended to back office systems such as valuation, administration and projection systems. It also makes sense that if the data standards efforts affect data and systems used by actuaries then actuaries should be playing a more active role in the development of these standards.

The major data standards initiatives do not appear to yet encompass standards for table type data, that is, anything that comes in the form of an "ultimate" or "select and ultimate" table such as mortality tables, lapse rates, premium rate tables, and cost of insurance charges. Being the primary users and creators of such data this is certainly an area where the actuarial profession can play a leading role.

With these goals in mind the Computer Science Section (CSS) of the Society of Actuaries (SOA) is putting forth an initiative to encourage the standardization of table data in the form of aggregate or select and ultimate tables. This includes tables for mortality, morbidity, mortality projection rates, lapse rates, premium rates, cost of insurance charges, surrender charges and other types of termination rates and age and/or duration dependant product feature data. This effort is a work in progress and the CSS is continually seeking feedback to ensure this effort fulfills the goals of users. The CSS is also looking for suggestions on how to manage the evolution and development of the standard.

This effort extends a prior project of the CSS to construct an Actuarial Tables Database called "TableManager" (described in detail below). It is the goal of the CSS to encourage software developers, insurance companies, pension consulting firms and other insurance and pension data standards initiatives to use the standard ASCII table format of the TableManager application. By using the standard format, age and duration dependant tables may be easily interchanged between different systems. This should result in significant workflow improvements in all back office departments as table data accounts for a large proportion of the data transferred across or shared between two different systems. Users may also benefit from an existing base of over 630 available tables including standard annuitant, population, insurance and regulatory table data from more than 15 countries. Additionally there is a binary database format that may be of interest to some developers but the thrust of this paper is to support the use of the ASCII table format.

This paper discusses the background of the initiative to support the adoption of the ASCII table format of TableManager, the reasons to adopt this standard and how to integrate it into insurance and pension systems. Also included with this White Paper is a survey requesting your input on your implementation plans and suggestions for future development of this standard.

It must be noted that the promotion of data standards is in no way intended to enforce a standard but rather to encourage efficient use of actuarial tables.

Background

Based on a proposal tabled in 1994 by Steve Strommen, the Computer Science Section (CSS) supported a project to build an Actuarial Tables Database. The proposal was to construct a database containing actuarial tables, an executable program to manage the database and a collection of computer source code to access and interface with the database, in various computer languages. The first version of this software, called TableManager, was made available in the summer of 1995 on Actuaries Online.

As of September 1998, over 630 "standard" tables from around the world have been converted to the SOA table format and are available from the TableManager application. This application allows users to view, edit, add and modify these tables, primarily mortality tables but also morbidity and mortality projection tables. Individuals may also export tables to ASCII file format or to Lotus spreadsheet format. All tables used by TableManager adhere to the same standard format.

Additionally, source code in C++ and an APL workspace are available to help developers integrate the database, ASCII tables and table formats into their own applications. The source code is available for free as part of the TableManager application.

The current goal of the CSS is to make SOA members, software developers and others in the industry aware of the existence of the Actuarial Tables Database, the associated table formats, the TableManager application and application programming interface (API) source code. The CSS also wishes to encourage software developers, insurance companies and pension consulting firms to incorporate the standard ASCII format into software systems that use or create age or duration dependant tables.

Why use TableManager and the SOA ASCII Table Format?

TableManager and the SOA ASCII Table Format are a unique non-proprietary development effort resulting from input by a large number of actuaries. The standard table formats are a result of extensive thought and discussion among actuaries with a broad background in computer issues. The result is an open standard with no hidden agenda and is fully supported by the SOA Computer Science Section.

The case for standard table formats is discussed at length in the overview section. In summary, it allows developers and users to quickly and efficiently interchange tables into any SOA Table Format Compliant system. This will improve efficiency, reduce errors and provide users with greater flexibility in their jobs.

TableManager also provides one location for over 630 "standard" actuarial tables. It is the most comprehensive source of freely available tables. The standard format is also a template for how companies may create electronic versions of their own proprietary actuarial tables. Any SOA Table Format Compliant system would then be able to easily incorporate these tables.

Additionally, integration and use of the tables into current systems is facilitated by free source code in C++ and APL. This source code shows developers how to manipulate the databases and tables and will allow them to quickly integrate this format into other computer systems.

Finally, open standards are a must for our industry and related professions to extract maximum value from the tools we rely on the most: our computer systems.

Who Should use the SOA ASCII Table Format and TableManager ?

Individuals, insurance companies, pension consulting firms and software developers should use the SOA ASCII Table Format and TableManager. The TableManager application will provide a source of over 630 actuarial tables; in itself the TableManager is a comprehensive resource. At a minimum, users can extract any table to suit their own purpose or to use in other applications. TableManager may also be used to maintain and edit, add or modify actuarial tables.

Additionally, software developers should consider using the SOA ASCII Table Format as the standard format for all age or duration dependant tables. If developers use this format and write the layer of code to manipulate these ASCII tables (or use source code from the provided APIs) then it is trivial for users to interchange actuarial tables. This will be of large benefit to the industry in general and the actuarial profession in particular.

How To use the SOA ASCII Table Format and TableManager

First, interested parties should download the TableManager application and become familiar with the application and supporting documentation. Information on downloading the TableManager application is available in the Further Information section.

Once comfortable with the application itself, the ASCII Table Format should be understood (see Appendix A for the definition and example) and, finally, for interested developers the source code should be studied as an example of how to develop classes or workspaces to manipulate the tables.

Software developers who create systems employing the SOA ASCII Table Format for all actuarial tables will be referred to as SOA Table Format Compliant (see the glossary for full definition). It is the primary goal of the CSS to encourage compliance - at least at this level.

Software developers who create systems employing the database format of the TableManager application for all actuarial tables will be referred to as SOA Database Format Compliant (see the glossary for full definition). The CSS would also like to encourage compliance at this level. However, the CSS considers the former level of compliance to have higher priority. Ideally the CSS would like to see vendors develop applications with this further level of compliance and to make use of the provided APIs.

The CSS will produce a list of vendors (and their relevant systems) who claim to be compliant with the standard and the level of compliance - SOA Table Format Compliant and/or SOA Database Format Compliant. This list will be distributed widely amongst the insurance and pension industries and also placed on the CSS pages of the SOA Web site. Initially, the CSS will also compile a list of those vendors who intend to be standards compliant in the near future, based on the answers to the questions in the survey attached to this white paper. This list will also be posted to the SOA Web site.

Initially, then standards compliance will be based entirely on the "honor system" - vendors who claim to be compliant will be included on the list of compliant companies. The CSS Council will then set up a system to receive user comments on the use of such systems as they relate to the standards and post these to the SOA Web site. This could possibly be done through a Web based bulletin board where users can post their own comments, and vendors their responses. Suggestions for procedures for alternative, independent or more comprehensive verification of standards compliance should be included with your survey response.

Current Known Limitations

As with any standard format there are limitations and the following have been identified with the SOA Table Formats and the TableManager application:

1. Does not support parametrically defined tables. Parametrically tables are tables that are defined by formula as opposed to static data.
2. Integer ages. The TableManager application currently only supports integer ages separated by one year. The TableManager application does not support interpolation of ages and, as a result, for example, quinquennial ages are not supported. However, the table format itself can be used and custom applications can be written based on the standard format to handle any required interpolation.
3. Proprietary tables are not covered. The databases do not contain proprietary tables nor are they expected to. However, proprietary tables are encouraged to be available in the standard format as it will make integration in SOA Table Format Compliant applications simple.
4. TableManager only runs on Windows. While TableManager only runs on Windows, the ASCII format of the tables will integrate with computer systems running in any language and on any operation system including UNIX and Mac. Additionally, if an individual or group of developers wish to create a version of TableManager for non-Windows operating systems, the final version can be presented and reviewed by the CSS for approval and release.
5. The tables included in TableManager have not been formally approved or checked by any group or organization. Rather they have been submitted on a voluntary basis by interested SOA members. If you feel strongly that these tables should go through a formal review process and be "blessed" by credible organizations (such as actuarial organizations of the countries to which tables apply) you should indicate this on your survey response and possibly also make suggestions for procedures, responsibilities and funding of such activity.

Future of the Standard

The CSS will be actively lobbying software developers, insurance companies and pension consulting firms to incorporate the SOA ASCII Table Format for all relevant application development. Insurance industry standards bodies will also be contacted to engage the SOA Table Format as the standard for age and duration dependant tables.

The CSS will also evaluate all suggestions and will determine if there is reason to further develop and enhance TableManager and to provide additional interfaces to TableManager data, such as a COM interface. The CSS is interested in receiving input from vendors, companies and standards organizations on how the standards should be developed in the future. Suggestions you might have for future development and the process for requesting and approving changes to the standard should be detailed in the attached survey.

The CSS and SOA will also be inviting other actuarial groups to participate and comment on this standard including the CAS, AAA, ASPA, CIA and CCA. Other international actuarial bodies will also be exposed to this standard and will be requested to provide comment and feedback.

The current version of the table formats will be referred to as Version 1.0. However, this is a work in progress and based on user feedback the format may be improved and enhanced over time.

Company and Vendor Support

The following companies have already expressed interest, have plans to develop applications or have already developed applications that are SOA Table Format Compliant:

Add disclaimer here.

| Company | URL | Status |
|---------------------------|--|--|
| Canada Life | www.canadalife.com | Single project is compliant. |
| Hutchison Avenue Software | www.hasc.com | Actuarial Kit and applications currently compliant (Java and C). |
| Northwestern Mutual Life | www.northwesternmutual.com | Single project is compliant. |
| TAS | www.towers.com/tas/ | Plan on integrating in 1999 |

In addition the standard format, TableManager and this White Paper are being sent to ACORD for consideration as a table data standard to be incorporated into the OLiFE standard at some future date.

For the most up to date listing of support please visit the SOA web page under the Computer Science Section.

DISCLAIMER: The Society of Actuaries makes no warranties as to the services of the companies and vendors listed above. The listing of them does not constitute knowledge, endorsement or approval by the Society of Actuaries of them, nor does the Society of Actuaries accept any liability for the services or work of the companies and vendors. Individuals using these companies and vendors do so at their own risk, and the companies and vendors shall remain individually responsible for any actions, statements or services provided to anyone who retains them. Because the Society of Actuaries is committed to adhering strictly to United States antitrust, copyright, trademark, securities and other federal statutes, as well as state common laws covering libel, slander, defamation, false advertising, invasions of privacy and violations of the rights of publicity, the Society of Actuaries strongly discourages users of the services of the companies and vendors from hiring them for anything that: (1) sets or controls prices or terms of products or services and the manners in which products or services are sold; (2) violates the proprietary or personal rights of others; or (3) constitutes a misleading advertisement. YOUR USE OF THE SOCIETY OF ACTUARIES' DATA STANDARDS IS ACKNOWLEDGMENT OF YOUR AGREEMENT WITH THE ABOVE.

Glossary

SOA Table Format Compliant - Indicates that a software application uses the ASCII table format as specified by the SOA TableManager program. The program will then be able to interchange any table and continue to correctly operate. All table types (example: aggregate and select and ultimate) must be supported to be compliant.

SOA Database Format Compliant - Indicates that a software application uses the SOA TableManager database format in addition to the table format. The application must be capable of interchanging any database in the SOA format and still function correctly. Although this is a higher level of compliance than being SOA Table Format Compliant, it is not the current goal of the Computer Science Section.

Further Information

Where to get the Table Manager program:

1. The SOA Web site (www.soa.org) under the Software and Technology Library section. The current link to download the TableManager Version 1.1 is www.soa.org/library/tblmgr11.zip.
2. The Canadian Institute of Actuaries Web site (www.actuaries.ca) with a current direct link at www.actuaries.ca/Database/indexeng.htm.

This white paper may be found on the SOA Web site (www.soa.org) under the Computer Science Special Interest Section.

Additional References to the SOA Tables Database Project:

1. "Some Thoughts on an Actuarial Tables Database", Stephen J. Strommen, CompAct, September 1994
2. "Council Approves OnLine Mortality Tables Database", Stephen J. Strommen, CompAct, February 1995
3. "An Update on the Mortality Tables Database", Stephen J. Strommen, CompAct, October 1995
4. "Mortality Table Database Update", Stephen J. Strommen, CompAct, January 1996

Who to contact:

Any member of the Society of Actuaries Computer Science Section Council will be able to forward any questions or inquiries to the correct person. You can e-mail the Section Council at cssc@ajug.org.

A current list of the Computer Science Section councilors is available on the SOA Web site.

Also feel free to contact Stephen J. Strommen (stevestrommen@northwesternmutual.com) or Scott Parkinson (scott@hasc.com) with questions.

Appendix A - Description of the SOA ASCII Table Standard

Required for all tables

| | |
|--------------------------|--|
| Table name | Any character string up to 49 characters (+) |
| Table number | An integer used for table lookup, unique to each table |
| Table type | Select or Ultimate |
| Contributor | Name of the person who contributed the table |
| Minimum age | Lowest tabulated age |
| Maximum age | Highest tabulated age |
| Select period (*) | Number of years in the select period |
| Maximum select age (*) | Highest age for which select values are supplied |
| Number of decimal places | Number of digits to which values are rounded |
| Table values | The numeric values |

(+) Table names can exceed 49 characters, but any characters beyond 49 are truncated in lists of table names.

(*) The Select period and Maximum select age are only required for select tables.

Optional additional information

| | |
|---------------------|---|
| Source of data | Where was the underlying experience data obtained? |
| Volume of data | What was the volume of underlying experience data? |
| Observation period | What time period was observed? |
| Unit of observation | e.g. number of lives, amount of insurance |
| Construction method | How was the table constructed from the experience data? |
| Published reference | Is the table published somewhere? |
| Comments | Any miscellaneous comments about the table. |

Example Table:

Here is a simple example of a select and ultimate table laid out as ASCII text. An explanation follows.

Table name: Some table

Table number: 12345

Table type: Select

Contributor: Somebody

Comments: This is a sample table.

The values don't really mean anything.

Minimum age: 0

Maximum age: 10

Select period: 3

Maximum select age: 5

Number of decimal places: 5

Table values:

| | 1 | 2 | 3 | 4 | |
|----|---------|---------|---------|---------|----|
| 0 | 0.00010 | 0.00012 | 0.00015 | 0.00020 | 3 |
| 1 | 0.00011 | 0.00014 | 0.00018 | 0.00028 | 4 |
| 2 | 0.00012 | 0.00015 | 0.00025 | 0.00037 | 5 |
| 3 | 0.00014 | 0.00024 | 0.00036 | 0.00047 | 6 |
| 4 | 0.00023 | 0.00035 | 0.00046 | 0.00058 | 7 |
| 5 | 0.00030 | 0.00045 | 0.00056 | 0.00070 | 8 |
| 9 | | | | 0.00080 | 9 |
| 10 | | | 0.00090 | | 10 |

There are two main parts to the layout of the table; the description section and the table values section.

In the description section, each kind of descriptive data is preceded by a specific phrase beginning on the left hand margin and terminated by a colon. The specific phrases that are used must be spelled exactly as shown in the list under the file formats topic. Note that the documentation on any topic containing descriptive text can extend across more than one line, as is shown above for the Comments topic.

The table values section begins with the line "Table values:". For select tables (but not for aggregate tables), this is followed by a line containing duration numbers as column headings. Subsequent lines contain an age followed by one or more table values.

Note that all data is space delimited (as opposed to Tab delimited).

Further examples can be produced using the TableManager program to further understand the ASCII standard format.

Appendix B - List of Other Insurance Industry Data Standards Initiatives

A. OLifE

A.1 Description of Standard

OLifE is an industry supported standard designed to integrate data between different software applications used in the insurance and financial services industries. The underlying technology of OLifE is OLE and the specifications are written assuming applications will be developed in a 32 bit Windows environment. The objects supported in OLifE define properties and methods needed to retrieve data. Such data objects include ;

- Party object. E.g. : policyholder, producer or carrier
- Holding object. E.g. : Policy - life, annuity and health - including product features, charges, investment options
- Investment product object
- Activity and group (e.g. : household) objects

The OLifE standard is currently focused on systems for contact management, illustrations, financial planning, electronic application and administration system data. Beyond the desktop the plan is to extend the OLifE standard to be applicable to distributed computing, legacy system support, service center support, Internet development and to apply to the insurance industry world wide.

A.2 Sponsoring Organization

ACORD (Agency-Company Organization for Research and Development) is a non-profit association dedicated to the development of insurance industry standards. Standards development is managed by ACORD through program management, technical support, change process, voting process and supporting services.

A.3 Participating Vendors

•Beacon Software Development Company •Benefit Technology •Borealis Corp. •Business Operations & Software Solutions Pty. Ltd. (Australia) •Cologne Reinsurance Company of Australia Ltd (COMPASS) •Coss Development Corporation •CSC •Cybertek •E-Z Data •ECTA •Emerging Information Systems (Canada) •FastForms •FDP Corp. •Financial Profiles •Finansys Inc. •FIPSCO •FM Systems (Ireland) •Genelco •Genesis Development Corporation •IBM (Canada) •Impact Technologies •Insurance Systems Laboratories (Japan) •Insurance Technologies •Intuitive Systems Limited (England) •Investment Data Technologies Pty. Ltd. (Australia) •Kettley Publishing •KinetiSoft, Inc. •Life Research Pty. Ltd. (Australia) •LifeLink Corp. •Lifeweb L.L.C. •Microsoft •ONTOS Inc. •Pacific Financial Systems (Australia) •Philibert Software Group •Plantech Consulting Group (Australia) •Protax Financial Services Pty. Ltd. (Australia) •Rolling Thunder Computing, Inc. •Sealcorp Holdings Ltd. (Australia) •Select Tech Inc. •Softbridge, Inc. •Solcorp (Canada) •Spidertek •Sterling Wentworth •Sun Microsystems •Swiss Re Life & Health Australia Ltd. (Magnum) •Sybase •Sybiz West Pty Ltd (Australia) •System Innovations •The National Underwriter Company •Tillinghast-Towers Perrin •United Systems

A.4 Participating Insurance Companies

•AEGON USA •Co-Operators •Lincoln National Life •Manulife Financial (Canada) •MetLife •Mutual of New York •Mutual of Omaha •National Mutual Financial Services (Australia) •NY Life •Pacific Life •Phoenix Home Life Mutual Insurance Company •Principal Financial Group •Provident Mutual Life Insurance Company •Prudential •State Farm Insurance •SunLife Assurance Co. of Canada •The Co-Operators Insurance (Canada) •The Equitable •Tower Life Australia

A.5 Contact Information

Contact : Tana Sabatino, OLifE Group Manager
 Phone : 800-444-3341 x440 (from USA)
 914-620-1700 x440 (from outside USA)
 Fax : 914-620-3640
 E-mail : tsabatino@acord.org , Web Site : www.acord.com

B. OMG Financial Domain Task Force (FDTF)*B.1 Description of Standard*

FDTF is an interface standard for the financial services industry focusing on seven generic management facilities;

- Product
- Agreement
- Party
- Financial transaction
- Portfolio
- Investment pool
- Asset and liability instance

B.2 Sponsoring Organization

OMG (Object Management Group) is the world's largest software consortium whose objective is to standardize interfaces and not functionality. They use a consensus based approach and develop standards to be open, realistic, vendor-neutral and international.

B.3 Membership

OMG's membership consists of several hundred companies from the software industry and many other industries, including financial services. For a complete listing of members by type see OMG's Web site at www.omg.org.

B.4 Contact Information

You can find the appropriate contacts at the following address: <http://www.omg.org/omg/contacts/directory.html>

C. Insurance Application Architecture*C.1 Description of Standard*

Proprietary standard developed by IBM focused on developing a high level business model for all lines of insurance including models for;

- Client information integration solution
- Insurance data warehouse
- InsureAgent

C.2 Sponsoring Organization

IBM

C.3 Participating Vendors

IBM

C.4 Contact Information

More information can be found at the following address; <http://www.insurance.ibm.com/insur/iaa/iaa.html>
Or e-mail : ask_iaa@vnet.ibm.com

Appendix C - Survey on Implementation Intentions and Future Development Procedures

1. General

Name : _____

Company : _____

Position : _____

Contact Details : _____

Address: _____

Phone: _____

Fax: _____

Email: _____

Company Type : Vendor

- Insurance Company
- Pension Consulting Firm
- Standards Organization
- Other

2. Willingness and Plans for Standards Implementation

(Questions for Vendor or Company Developing Internal Applications)

a) Will you be likely to adopt this Standard ?

b) At what level : Table Format Compliant
 Database Format Compliant
 Both

c) If Yes to a) ...

i. What system or systems you develop will become standards compliant ?

ii. When do you think these systems will first become standards compliant ?

d) If No to a) ...

i. Why ?

ii. What would make you change your mind and adopt a table standard ?

3. Changes to Standards and Standards Development Procedures

(Questions for all Respondents)

This is version 1.00 of the table data standard. As with other standards this standard will progress over time.

- a) Do you have any suggestions for later versions of the standard ?
Eg : disability termination rates, monthly values, any other table features not included herein.
Please List
- b) How do you think suggestions for changes should be incorporated into the standards? (Yes or No)
- i. Submission to CSS Council for approval ?
 - ii. Voting by SOA CSS Members ?
 - iii. Voting by SOA Members ?
 - iv. Voting by all interested parties in insurance industry ?
 - v. Voting by special sub-committee appointed by SOA CSS Council?
 - vi. Voting by special sub-committee nominated/elected by interested parties in insurance industry?
 - vii. Outsource process to existing/established standards organization such as ACORD?
 - viii. Other ? Please describe.
- c) How should people make requests for changes ?
- i. E-mail / mail to SOA CSS Council ?
 - ii. Bulletin Board on SOA Web site ?
 - iii. Other ? Please Specify
- d) How do you think suggested changes should be communicated to all interested parties ?
- i. SOA Web site ?
 - ii. Mailings - Paper or E-mail ?
 - iii. Other ?
- e) Do you see a need for a Data Standards Bulletin Board and/or Discussion Group on the SOA Web site?
- f) What topics should be covered in such Bulletin Boards or Discussion Groups?

4. Standards Verification Procedures

(Questions for all Respondents)

- a) Do you agree with the the overall approach of identifying systems and companies as being "compliant" with the standard, and being able to publicize this fact?
- b) If you said "no" to 4.a) above, why is that? Please explain.
- c) Are you happy with the "honor system" of standards compliance?
- d) Do you see a need for independent verification via a more formalized process?
- e) If so, do you have suggestions for how such a process might operate and how it might be funded?

- f) Do you think the standard tables in the tables database need to undergo a formal review process and be "approved" by a credible organization(s)?
- g) If yes to f)..
 - i. Who should have responsibility for approving the tables?
 - ii. What should be included in the review process?
 - iii. How should such a review process be funded?

5. Other Standards Initiatives

(Questions for all Respondents)

- a) Is your company a member of, or somehow involved or participating, in any other standards initiative?
- b) If yes, what standards initiatives are you involved in?
- c) What systems are, or, will be, compliant with these standard(s)?
- d) Are you aware of any other table standards initiatives in the financial services industry?
- e) If yes, what is the sponsoring organization and the relevant standard?
- f) Would you like to see the table standard as described in this White Paper incorporated into other insurance industry data standards efforts?
- g) If so, which ones ?

6. TableManager Platforms

While the proposed standard table data ASCII format will integrate with with computer systems running in any language and on any operating system, as noted in the White Paper "TableManager" runs only in Windows. Do you think TableManager should be developed to run on other platforms, and if so, which ones?

7. General Comments

(Questions for all Respondents)

Please list in this section any other comments or suggestions you have in relation to the proposed data standards.