Integrating Best of Breed Outage Management Systems with Mobile Data Systems

Donald Shaw
Partner
ExtenSys Inc.
31 Plymbridge Crescent
North York, ON M2P 1P4
Canada
Telephone: (416) 481-1546
Fax: (416) 481-6714
E-Mail: dhs@extensys.com

Christine Hoffmann
Senior Business Analyst
Baltimore Gas & Electric
7309 Windsor Mill Road
Baltimore, MD 21244
Telephone: (410)597-7243
Fax: (410)597-6310
E-Mail:
Christy.Hoffman@bge.com

Abstract

Both Outage Management Systems and Mobile Data Systems provide great value to an electric utility. A tight integration between the two multiplies the value of both.

This paper discusses the challenges involved in such an integration including:

How do you choose the data to be passed between the two systems?

- What do the various users need to do their work? OMS operators, MDS dispatchers, Field crews, etc.
- How do the users of the two systems keep a consistent view of the outstanding work?
- What are the source and destination of the data?
- How to access and store the data? Via API, database access, etc.

How do you manage apparent incompatibilities between the object models of the two systems?

- An Outage on the OMS is not always the same as a Work Order on the MDS
- Both systems may be able to manage crews. Which system is the master? Should the other system track the crews?
- Relationships between objects are not always the same. Can more than one crew work on a Work Order or an Outage?

How do you manage the special needs of each system?

- Transmitting data can be expensive. Where should change detection be used?
- Where are the data editing rules enforced?

Examples from a variety of projects illustrate the discussion points and highlight different approaches.

BIOGRAPHICAL INFORMATION

Donald Shaw
Partner
ExtenSys Inc.

Specific Responsibilities

Co-founded ExtenSys in 1993 and has been working with Outage Management Systems (OMS) since 1993, and Mobile Data Systems (MDS) since 1997. Responsible for technical leadership in client projects, in particular for specifying, building and installing interfaces between near-real-time systems. Several major US electric utilities are achieving benefits from the integration of OMS and MDS due to Donald's work.

Past Experience

Prior to forming ExtenSys, Donald worked for over 15 years for CGI/Gellman,Hayward on a variety of real-time industrial system projects in a number of roles including specification, design and software development. In the early 1990's Donald worked on the North York Hydro project that was one of the first instances of a product-based distribution management system.

Educational Information

BA – Engineering, Cambridge University

BIOGRAPHICAL INFORMATION

Christine Hoffmann
Senior Business Analyst
Baltimore Gas & Electric

Specific Responsibilities

Joined Baltimore Gas & Electric (BGE), a member of the Constellation Energy Group, in April 2004; however, she has been working at BGE since March 2002. Christy has been working on the Mobile Dispatch Project since its inception. She was responsible for the requirement specifications for the interface between the Mobile Dispatch System (MDS) and the Outage Management System. She is currently responsible for a multitude of tasks to support MDS on a day-to-day basis. Her tasks include functional analysis, coding, testing, and installation.

Past Experience

Prior to joining BGE, Christy worked for Applied Information Sciences (AIS), a software consultant firm. While at AIS, Christy worked on a variety of projects as a business analyst.

Educational Information

BS – Computer Science and Mathematics, Towson University

Background

This paper presents our experiences working on the interface between MDS and OMS using a variety of best-of-breed products. We hope that utility staff considering doing an interface can learn from this. Those who have done this already can use this paper as a gauge to see if their experience is typical.

Our experience is presented as a story representing an amalgamation of our project experience. The story poses the question: "We have two best-of-breed systems. How do we combine them to make the most of their capabilities?"

In this story, an electrical distribution utility is using a best-of-breed approach. They have an OMS in service, and know its capabilities well, and have selected an MDS to dispatch work to crews in the field. The MDS is to be used to dispatch multiple types of work to crews with varying capabilities. This work is to be supplied to the MDS from a number of different systems including a Work management system, as well as the OMS. This paper describes the integration between the OMS and MDS only.

They understand the value an OMS provides by attaching customer calls to the appropriate devices in the network model and using the call pattern to predict the location of outages. The predictions are improved when the state of the network devices are kept up-to-date, using SCADA information and reports from field crews.

They also understand the value of an MDS presenting accurate information to the field crews, and allowing the field crews to return information gleaned in the field to the OMS operators quickly.

The Story

The story begins with the initial meeting between Ms. FBA, the Utility's Functional Business Analyst, and Mr. IE, the OMS vendor's Interface Engineer.

IE: "I see you've done your functional analysis and you know your data requirements, but I need to know how to access the MDS."

FBA: "They are final stages of implementing an interface that provides access to their internal messaging bus."

IE: "What form does this interface take?"

FBA: "They use a middleware product to exchange messages formatted in XML."

IE: "We've done a couple of interfaces to other vendors' MDS's, but XML was not an option when we did them. We have an XML interface to our messaging bus, I wonder if we can use that, using the middleware's connector technology to transform the data."

FBA: “That sounds interesting, what else do you need to know?”

IE: “As you know, the connection between the MDS and the Mobile Data Terminals (MDT) in the field crew’s vehicles has a limited bandwidth. This means that the MDS must be careful to send only the data that has changed. We must ensure that the way the interface transmits data does not defeat this.”

FBA: “Yes the MDS vendor warned us about this. Any data that is sent via the interface that is viewable on an MDT is transmitted as soon as possible.”

IE: “Hmm. This means that the interface must not send data that has not changed. This implies that the interface needs to save the data previously transmitted so that it can do change detection. This kind of persistence of data is awkward to do in a middleware connector, but I’ll think about it.”

FBA: “Now we’ve looked at the way the two systems communicate, we should look at the way the systems will look to the users.”

IE: “Why not give me an overview of a simple scenario?”

FBA: “How about working an outage job?”

IE: “Yes, that’ll give me a good feel of the interaction.”

FBA: “A customer notices that their power is out, and calls us. The Customer Service Representative enters the information on the Customer Information System. The call data is transferred to the OMS and is then analyzed to predict where the outage is. Once this analysis is completed a new probable outage appears on the OMS Job List. This should trigger the creation of a Work Order on the MDS.”

IE: “This brings me to my first question. When does the interface know to create the Work Order?”

FBA: “Why not as soon as it appears on the Job List?”

IE: “Well, one of the advantages of an OMS is that it analyzes multiple customer calls and groups them according to the network topology, often re-predicting from the original estimation of the problem. For example, two apparent service outages may group up to a common lateral fuse. This grouping can occur quite quickly, especially during the early stages of a storm.”

FBA: “I still don’t see what the difficulty is.”

IE: “Sorry, I didn’t fully explain. If the Work Orders are created immediately, and then some grouping occurs, at least one of them will have to be cancelled on the MDS. This can cause unnecessary data transmission and waste resources on the MDS, not to mention possible confusion for the MDS users. Therefore it is preferable to allow some time before Work Order

creation."

FBA: "Now I see. What are the alternatives?"

IE: "In the past, we have triggered Work Order creation using two methods. First, by waiting fixed length of time, say 20 seconds, and second waiting for an OMS user to manually intervene. I, personally, prefer the second alternative, as it allows OMS users to manage the load on the MDS."

FBA: "Yes, we have seen that method used at another of your customer's site. As I recall, the method chosen was to create the Work Order when the OMS user acknowledged the Job on the OMS. Is there another way?"

IE: "Yes, we can add a 'Send to MDS' button on the OMS Job List. However, since you have had the OMS in service for some time, and your users are familiar with acknowledging the Jobs, I would recommend going that route."

FBA: "Ok, back to the scenario. The OMS operator acknowledges the Job, and an MDS Work Order is created. The Work Order is assigned to a field crew by the MDS dispatcher, and the status of the Job on the OMS changes to assigned."

IE: "So the interface needs to listen for Work Order status changes, and needs to keep the OMS crew database up-to-date. This brings up a perennial problem. We have two systems that control the state of the work, and both of them have crew management capabilities. We need to prevent the systems getting out-of-sync from one another. Usually, that means deciding which system is the master for such data. The roles can change depending on the state of the work. In particular, which system should be the source of the crew data?"

FBA: "Your other customer chose the MDS, and disabled some of the crew functions on the OMS. We thought that we'd do the same."

IE: "That makes sense. Ok, the MDS is where crews are created and the interface has to keep the OMS crew database up-to-date."

FBA: "Ok, back to the scenario again. The field crew starts driving to the Work Order location and presses the 'En-route' button on the MDT screen. When the crew reaches the location, they press the 'On-site' button. The state of the Work Order changes to reflect these state changes on the MDS and on the OMS. The crew then surveys the site to determine where the problem is. Assuming the prediction was correct, the crew enters the cause of the problem and the estimated time to restore the power."

IE: "Yes, this data is entered into the OMS database, making it available to the CIS so that if a customer who is affected by the outage calls, the Customer Service Representative can say that a crew is working on the problem, and when the power is expected to be back on."

FBA: "We need to do a little more, however. When the crew confirms the location of the

problem, we want the OMS to know where the exact location of the outage is, so that it doesn't try to group it with other problems."

IE: "Ah, so the interface needs to be able to change the status of the network model. Our previous customers have had OMS users do any changes to the network model. OMS users often prefer to do all changes to the status manually."

FBA: "Yes, the team here has had that discussion. The compromise that we came to was that confirmations from the field would be automatically performed if the problem was either with the transformer directly connect with the customer, or with the service to the customer's premise."

IE: "That's a good compromise. Don't interrupt the OMS user for small problems, but get them involved when there are larger implications. Still, operating the model is a requirement we have not needed to meet before."

FBA: "On with the scenario. The crew fixes the problem, and enters data describing what they have done, including the actual time that the power was restored."

IE: "Good, having the actual time of restoration is much better that noting the system time when the message arrives at the OMS. The message can take some time to arrive, especially if communications between the MDT and the base station are interrupted. By saving the actual restoration time, your outage statistics are more accurate."

FBA: "Yes, that's the point. Anyway, once the completion information is returned to the OMS, and if the field crew indicates that no follow up work needs to be done, the job's status is changed to complete on the OMS. This causes the outage to be removed from the active jobs on the OMS Job List."

IE: "As in the outage confirmation, you have gone further than some of our other customers. They need the job to be reviewed by an OMS user before it is completed."

FBA: "Yes, and we want the same rule to apply, this should only be done for customer transformer or service problems."

IE: "Well, that's given me a lot to think about. I'll have to do a fair bit of thinking about how we're going to do the interface. In addition, I need to study your detailed data requirements to see how to access the data the MDS needs from in the OMS, and how to save the data coming back from the MDS."

FBA: "Ok. I'll see you in a week for another meeting, but call or e-mail me any time you have questions."

IE: "That sounds good. Call me if you think of anything important I've missed, or if you have any questions."

IE goes back to base to do his analysis with the rest of the vendor's team. He returns a week later for the next meeting.

FBA: "Well, we've both had a busy week. What do you want to talk about first?"

IE: "I have a few questions about the data requirements and some comments about some of the implications of what you would like to do."

FBA: "Go ahead."

IE: "To give you some background, there are three main sources of the data from the OMS. First of all, when an OMS job is created or changed, an asynchronous message is broadcast to the interface. This message contains most of the data you would like to send to the field crews. Next, data can be obtained from the database directly by executing queries. Thirdly, some of the data is cached by server processes, and is accessed via the OMS API. Each of these has performance implications."

FBA: "For example?"

IE: "Data in the asynchronous message is essentially free, because the message has to be sent to the interface to signal a change in the job. Next, fetching data from a server cache is usually less of a performance hit than querying the database, but it does use up resources. In addition, each API call and database query uses resources, so we would like to keep them to a minimum. The interface can mitigate some of this by caching data that does not change often, but there may be other ways to avoid these calls or queries."

FBA: "We're all for keeping performance up. We know what can happen in a severe storm."

IE: "For instance, you want the field crews to have meter information. In your current configuration, that would require a query involving a join of three tables. In other projects, the meter information has been packaged with the customer call data. If we can convince our OMS teams to add that data to the customer call we can avoid the meter query each time the job changes."

FBA: "Ok, I'll talk to my team, you talk to yours."

IE: "More importantly, I see that you want to send a lot of customer call data to the field. In the past, we have only sent the first call's data. Generally, this data does not change very often, so that it can be cached in the interface. You would like to see not only the first call, but also the five most important calls. That can be a lot of data to fetch and sort each time the job changes, for example when a feeder is out during a storm. We are concerned about performance during heavily loaded conditions."

FBA: "The MDS dispatchers need to be able to assign the most important Work Orders first, and the crews would really like to see this data."

IE: “I agree that this data is valuable, but we may have to revisit this after doing some performance tests.”

FBA: “I’ve been doing some experimentation with our test OMS, and I think we’ve come up with a bit of a problem. The situation is that two customers on the same transformer call, and the field crew discover that it’s a service outage. Therefore we confirm this in the OMS and we end up with two jobs. The original job for the first caller, and a new job for the second caller.”

IE: “Yes, we consider this situation to be two service outages.”

FBA: “But we only want one Work Order on the MDS, as the same crew will fix both problems.”

IE: “Ok, we have a mismatch here in the object models between the OMS and MDS. Each model is entirely valid in the domain that the system addresses, but we need them to work together. This kind of problem is one of the trickiest when building an interface between complex systems. We’ve seen this particular mismatch before, and we have seen others.”

FBA: “Well, how are we going to fix it?”

IE: “What we need to do is to combine the information in the two OMS jobs and use this as the source of the data for the Work Order. In addition, we need to feed the information returned by the MDS for the Work Order to one or both of the OMS jobs. We need to define how to do this. In addition, the interface has to have a set of rules that defines when two or more jobs need to be combined.”

FBA: “The combination rule should be that multiple service outages on the same transformer should be combined.”

IE: “I like that rule. The easier it is to explain a rule, the easier it is to implement. How should the data be combined? In previous projects most of the data is extracted from the first caller’s job, but a few pieces of data are either summed or joined together.”

FBA: “What have you summed and what have you combined?”

IE: “All customer counts are summed, so the total number of calls, customers out and critical customers is correct. In addition, we combine the trouble codes. In your case we would combine all the customer calls into one list.”

FBA: “That sounds good to me. I’ll discuss it with the rest of the team and get back to you. Is there anything else we need to watch out for?”

IE: “Well, in another project we had the reverse problem. The MDS’s model only allowed one crew to be assigned to a Work Order. If more than one crew had to do the work, they created another Work Order linked to the original by a parent-child relationship. We needed to track the crew assignments on the OMS, which allows multiple crews assigned to a job. The interface

could detect the parent-child relationship, so assignments to the second Work Order were applied to the job for the original Work Order."

FBA: "We don't have that problem here. The MDS allows multiple crews on a Work Order."

After many meetings to refine the requirements, the construction of the interface starts with an agreed schedule and budget. At the end of the construction and functional testing, it's time for performance testing. After the first day, FBA and IE sit down to discuss the results.

IE: "Wow, that went well! The numbers were better than we specified."

FBA: "Mmm. Do you remember the hurricane we had in the fall?"

IE: "Yes, easily the worst storm you've had."

FBA: "Well, the steering committee has directed us to test the systems under the same conditions."

IE: "But that's much more than we anticipated."

FBA: "Well, as my boss said, there's no point in having two best of breed systems integrated when you have to disconnect the two systems when they are needed most."

IE: "He has a point. We'll need to get our thinking caps on."

FBA: "How do you suggest we proceed?"

IE: "We should test the interface to find out the heaviest load it can manage, and use instrumentation to determine where the bottlenecks are. Then we can either fix the bottlenecks or work our way around them."

After several iterations of optimization and testing, the interface still does not quite meet the new, more stringent, performance requirements. They meet again.

IE: "We're still not there yet. Have you any ideas?"

FBA: "Remember way back when we were discussing the customer data and you said that it might have an impact on performance?"

IE: "Yes we've measured the impact, and it's high, but your team decided that the data was necessary."

FBA: "Well, we've come up with a compromise. Would it be possible for us to turn off reading the customers when the load is very heavy?"

IE: “Boy, why didn’t I think of that? Since reading the customer data is part of the business logic in the interface, which is, by design, easy to configure, that’s a comparatively simple change. Lets give it a try.”

The compromise to optionally reduce functionality to boost performance during heavily loaded conditions is deemed to be acceptable. The interface is put into production. Everyone lives happily ever after, except during storms.

Conclusion

The OMS/MDS interface helps the two systems to do what they do best, gathering, analyzing, and presenting data to their user communities to help them to perform the task at hand. New information on the OMS can be transmitted to the field crews without any interruption to the OMS users. Similarly, information gathered in the field can be transmitted to the OMS without it having to be entered manually on the OMS, keeping the OMS data more current and more accurate.

Both the OMS and the MDS were configured to meet the needs of the users and the business logic used to process the data. Similarly, such an interface is needs to be configured to meet the data and logic needs of the two systems.

This paper illustrates the necessity of communicating the needs of the user community, and the compromises required when conflicting goals have to be resolved.