Using Archon, Part 2: Electricity Transportation Management

Jose Manuel Corera, Iberdrola
Inaki Laresgoiti, Labein
Nick R. Jennings, Queen Mary and Westfield College

THE ARCHON SOFTWARE FRAMEWORK INTEGRATED SEVEN HETEROGENEOUS AGENTS—SOME PREEXISTING SYSTEMS AND SOME CUSTOM-BUILT—INTO A FUNCTIONAL REAL-WORLD APPLICATION. THIS DAI APPROACH PROVIDES ECONOMY, ROBUSTNESS, RELIABILITY, AND A NATURAL REPRESENTATION OF THE DOMAIN.

ENERGY MANAGEMENT MONITORS and controls the cycle of generating, transporting, and distributing electrical energy to industrial and domestic customers. Generation transforms raw energy (for example, hydraulic, thermal, nuclear, and solar energy) into a more accessible form. Ideally, the consumption sites would be near these generation sites. However, various economic, social, and political factors make this often impossible. So, energy must be transported from its generation site to the consumer. To minimize losses during transportation, the electrical voltage is increased (to 132 kilovolts or more) before it is placed on a *transport network* and sent over many hundreds of kilometers. Finally, the voltage is lowered, and the electricity is delivered to the consumers using a *distribution network* that involves many kilometers of network (all below 132 kV) spread over a much smaller area.

Iberdrola is a Spanish electric utility with a generation capacity of 16,715 megawatts and a maximum demand of approximately 10,000 MW. Its transport network divides into three voltage levels and comprises 401 busbars (conductors for carrying current or connecting circuits), 296 lines, 294 transformers, 939 breakers, and 2,322 switches. A *remote transmission unit* at each of the 123 substations acts as an interface between the network and the North *Dispatching Control Room* in Bilbao, which controls the network. The DCR assimilates transport-network information (amounting to 25,000 data points) acquired by the RTUs. This information includes analog measurements and digital signals. However, during diagnosis, the operator focuses predominantly on the alarm messages corresponding to fault recorders, breakers, and protective relay operations.

To ensure that Iberdrola's transport network remains within the desired safety and economical constraints, Iberdrola uses a sophisticated data-acquisition system called Scada (*s*upervisory *c*ontrol *a*nd *d*ata *a*cquisition) and several conventional application programs that help the operator (the *control engineer*) to analyze it (these programs are primarily designed for normal operating conditions). Whenever an unexpected event occurs, the Scada system automatically sends hundreds of alarms to the DCR. Under these circumstances, the operator must rely on experiential knowledge to analyze the information, diagnose the situation, and take appropriate remedial actions to return the network to a safe state. To reduce the operators' cognitive load in such circumstances, and to help them make better decisions faster, Iberdrola decided to develop several decision-support systems. They then interconnected these systems and subsequently extended them using Archon distributed AI technology (see "Using Archon to Develop Real-World DAI Applications, Part 1," on p. 64).

Why use DAI techniques for this application?

When Iberdrola decided in 1988 to implement decision-support tools to ease the workload of their control engineers during disturbances, several technical factors influenced their

 0885-9000/96/$4.00

design choices. First, the control system itself was a proprietary product from a control systems supply company—Iberdrola considered it too risky and too difficult to embed the additional functionality directly in that system. Second, the state of the art for commercial systems in this domain was such that realization of the diverse support functions required a number of stand-alone systems.[1–4]

Consequently, Iberdrola built separate decision-support systems to assist with different aspects of the control engineer's job. The aspect that is most relevant to this discussion is the alarms-analysis expert system, which diagnosed faults produced in the network, based on the alarm messages that arrived at the DCR. These decision-support systems were unconnected, except that they retrieved information about the network (the current state of breakers, the activated protective relays, and the power flows and voltage measurements) from the same source (the control system's real-time database). To make this information available to the nonproprietary software products, Iberdrola had to develop several interfaces to the control system. As well as providing access, these interfaces could filter and preprocess network information.

By 1991, however, three important changes had occurred:

- The evolution of information technology hardware and software had significantly increased the quantity and quality of the data that could be acquired from the transport network;
- Improvements in local area network technology had made distributed computing commercially viable; and
- The prices of computers had decreased such that powerful machines were no longer prohibitively expensive.

Taken together, these changes meant that better and more powerful tools could be built to assist the control engineer.

In particular, Iberdrola wanted to be able to perform and dynamically monitor service restoration and to exploit new data sources such as chronological information or faster rate snapshots. However, they still needed tried-and-tested decision-support tools. So, Iberdrola decided to adopt a system-upgrading strategy that would enable the previously operational components to work in conjunction with the new functionality. They considered two means of realizing this strategy: extend the existing systems to cover the new features, or follow a distributed approach that expresses the new functions as distinct computational entities that could interact with the preexisting systems through a common distribution platform. Iberdrola chose the second option because they thought that it more effectively meets four basic criteria.

First, it permits reasoning based on information of different granularities. This application now needed to deal with two types of alarms: *chronological* and *nonchronological*. In chronological alarms, the time stamped coincides with the event's actual occurrence. In nonchronological alarms, the time stamped coincides with the time of acquisition by the control system (consequently, it is conditioned by the control system's polling mechanism). Because chronological alarms more accurately portray events in the network, they generally lead to a swifter diagnosis. However, chronological information has a low priority in Iberdrola's communication channels. So, when the channels are saturated (as can happen during a disturbance), their availability time is unpredictable. Therefore, Iberdrola decided to build a new alarm-analysis expert system that used chronological information and that could subsequently integrate its results with those of the preexisting system, rather than construct a monolithic system that received both types of data and that had to embody both types of diagnostic knowledge.

Service restoration presents a similar situation. This activity involves two types of information: *snapshots* (which provide a comprehensive picture of the current state of all the components in the network) and *alarm messages* (which show how the state of the components has changed over a period of time). Snapshots can be produced relatively quickly. Alarm messages might take several minutes for a large disturbance but are needed to indicate the type of fault from which the system must be restored. Rather than trying to place both types of information and reasoning in a single system, it seemed more natural to develop a service-restoration subsystem that dealt mainly with snapshots. Such a subsystem would receive the necessary high-level information about the faulty equipment from a diagnosis subsystem (rather than deal with the raw alarm messages itself).

Second, a distributed approach allows the system to include different network models. Some of the problem solvers needed to work on the Scada model of the network, while others needed the applications network model (a model that permits network equations to be solved and that takes into account the physical characteristics of all its components). Rather than trying to combine and harmonize these complex and disparate models at design time, Iberdrola decided that each subsystem should work on whichever model was most appropriate for its task. Then, the various components should be able to interact at runtime to resolve any inconsistencies that arise from their use of different network models.

Third, it enables the use of a number of different problem-solving paradigms. The diverse range of activities that this application required meant that there was no universally best problem-solving paradigm: procedural techniques were required for algorithmic calculations like connectivity (to know which component is connected to which other) and load-flow analysis (solution of the network equations), whereas symbolic reasoning based on heuristic search was the best approach for diagnosis. A distributed approach let us encode each component in the most appropriate method.

Fourth, it meets the application's performance criteria. Transportation management is a time-critical application. Because many different types of information can process in parallel, with only a small synchronization overhead, using a number of interconnected machines can improve the overall system's response time.

Having decided on a distributed approach, Iberdrola had to choose between using more conventional distributed-processing techniques or DAI techniques. They adopted the latter for these reasons:[5,6]

- *Economy*: The alarms-analysis expert system was already operational; however, it needed to incorporate new functions and treat new information. The estimated cost of modifying the extant system was significantly larger than that of implementing a new one. However, Iberdrola also judged that, because the new functions and data were so diverse, putting them in a single system would be extremely expensive. Therefore, it was more economical to build smaller systems, reuse the existing alarm-analysis expert system, and integrate them through a DAI framework. A DAI framework was needed because the interactions between these subsystems were both sophisticated and context-dependent. Therefore, runtime reasoning based on dynamic data was necessary.

- *Robustness*: Because the subsystems have overlapping domains of expertise, one subsystem's failure to produce an answer does not necessarily mean that no solution will be forthcoming; another system might be able to produce at least a partial solution. However, to flexibly achieve this backup functionality, the different problem-solving components must be intelligently coordinated—a task beyond present-generation distributed-processing systems.
- *Reliability*: The solutions of the overlapping systems can be cross-referenced so that the operator can obtain more reliable information. Again, however, this cross-referencing must be properly managed according to the prevailing circumstances, so it requires dynamic and flexible reasoning.
- *Natural representation of the domain*: A DAI approach accurately represents the way the control engineers work when a large disturbance occurs. They specialize their roles—one looks after restoration, another tries to diagnose the problem based on different sources of information, and so on. They then communicate relevant information to one another to ensure they are following a coherent course of action toward the overall objective of restoring the service.[7]

Specification of the agents

During normal working conditions, management of the network by the operator in the DCR consists mainly of topology changes (operation on breakers and switches), generation scheduling, and control of the energy interchange with other utilities.[8] However, during emergencies, management becomes considerably more difficult because of the large number of constraints to consider and the insufficient quality of the information that is available to make these decisions. Emergencies typically originate from a short circuit in a line, busbar, or transformer. They can be exacerbated by equipment malfunctioning (for example, a breaker failing to open) or subsequent overloads (a domino effect can cause one line to fail because of an overload; this in turn increases the load on neighboring lines so that they overload and subsequently fail, and so on). The situation can become even worse if power stations become disconnected, because this will cause an imbalance in the network's power.

Consequently, actions to restore service must be rapid and accurate, so that what starts as a relatively minor problem does not escalate into a major disaster. In these circumstances, the operator can perform mainly breaker operations, topology changes, and activation or deactivation of automatisms (various network components) and protective relays. Larger disturbances, however, might also require actions on power plants.

From this description of the control engineer's job, a top-down analysis identified that a comprehensive decision-support system should

- Detect disturbances; sometimes routine maintenance can trigger protective relays and breakers, and this should not be confused with a genuine disturbance.
- Determine the cause, location, and type of the disturbance, including determining if any equipment is permanently damaged.
- Analyze the network's situation once it arrives at a steady state.
- Prepare a restoration plan to return the network to its original operational state.

ACTIONS TO RESTORE SERVICE MUST BE RAPID AND ACCURATE, SO THAT WHAT STARTS AS A RELATIVELY MINOR PROBLEM DOES NOT ESCALATE INTO A MAJOR DISASTER.

Allying this top-down analysis with the bottom-up perspective of examining the extant systems, we decided to encapsulate as agents the alarms-analysis expert system and the interface to the control system. As we discussed previously, the availability of chronological alarm messages necessitated a new diagnosis system, which we decided to make available as an agent. Finally, Iberdrola always knew that information about the initial area out of service (the *blackout area*) could help constrain the search for the faulty equipment. However, they never deemed it cost-effective to develop a dedicated standalone system for this purpose, because they considered the original alarm-analysis expert system's performance satisfactory (if somewhat slow). But DAI technology made available from other agents much of the basic infrastructure to implement this functionality. So, they considered it economically viable to develop a system capable of producing this information. (In terms of the methodology we described in Part 1, p. 64, "Designing a multiagent community," this decision corresponds to providing additional functionality through the development of new systems).

The operational DAI system consists of seven agents running on five different machines (see Figure 1). This figure shows a small portion of the Iberdrola network, which contains four substations (Sestao, Sodupe, Erandio, and Achuri). Each substation's RTU sends information about the status of its busbars, breakers, and other electrical components to the DCR. In the DCR, the front-end computer collects this information and makes it available to the cooperating agents through the control system interface.

The seven agents are

- *Breakers and relays supervisor*. The BRS is the new alarms-analysis expert system. It detects a disturbance, determines the type of fault and its extent, generates an ordered list of fault hypotheses, validates hypotheses, and identifies malfunctioning equipment. To perform its analysis, it takes two types of inputs: chronological alarm messages, and snapshots of the network that give the status of every breaker and switch.
- *Alarms-analysis agent* (preexisting). This expert system handles nonchronological information. It pursues goals similar to those of the BRS; however, the quality of information it receives is inferior to that of the BRS. Although the alarm messages received by both systems relate to the same physical operations, those received by the AAA represent ±5 seconds accuracy, while those received by the BRS are exact. If the data is error-free, the BRS performs a better diagnosis than the AAA. However, if some of the chronological information is lost (a distinct possibility when the Scada system is busy), the BRS might perform worse than the AAA. Therefore, whenever incomplete or erroneous information exists (as happens in the most interesting cases), the

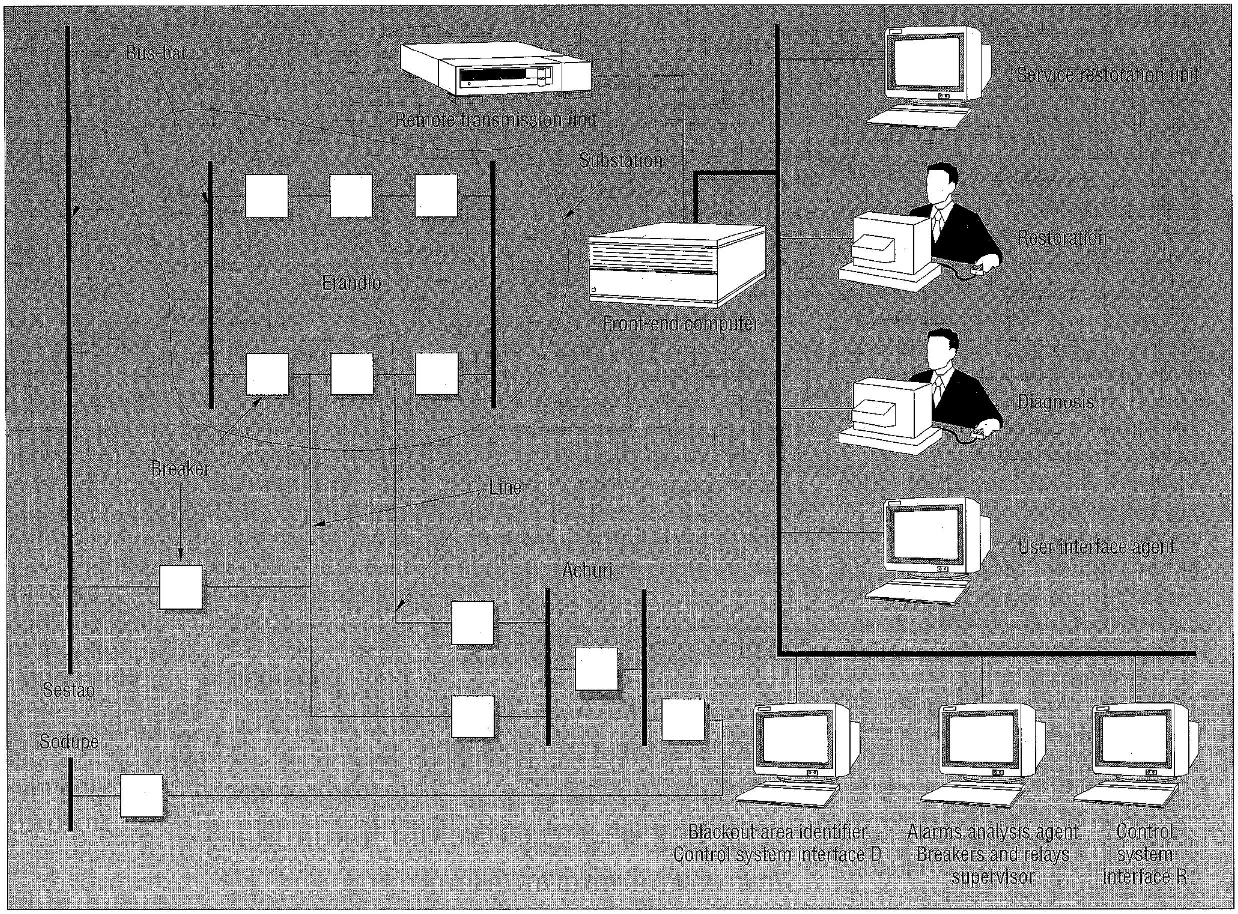

Figure 1. The transport network and Iberdrola's agents.

two systems must cooperate to make the overall system more robust and reliable.

- *Blackout area identifier.* When a fault occurs, the network's protective relays and breakers automatically try to isolate the minimum amount of equipment possible; in an ideal case only the faulty element would be isolated. The BAI identifies which network elements are initially out of service, because the actual element at fault must be in this region. It uses nonchronological alarm messages as its information source and cooperates with the BRS and the AAA to increase the efficiency of the overall diagnosis.
- *Service restoration agent.* The SRA devises a service-restoration plan to return the network to a steady state after a blackout has occurred. To do this, it takes into account the constraints imposed by the damaged equipment, as identified by the diagnosis agents.
- *User interface agent.* The UIA implements the interface between the users and the community of agents. Using a distributed windowing system, it presents the appropriate information on the consoles of the control engineers who are working on the system. (Figure 1 shows two such control engineers—one working on restoration activities and one working on diagnosis activities). It lets the engineers inspect the results produced by the diagnosis agents, view the alarms received, and browse through the log of analyzed disturbances. The engineers working on restoration can view, modify, simulate, or request plans.
- *Control system interface* (preexisting). The CSI acts as the application's front end to the control-system computers. It acquires and distributes network data to the other agents, interfaces to the conventional management-system application programs, and monitors the restoration to detect any unexpected deviations. It is split into two physical agents. The *CSI-D* primarily handles diagnosis. It detects disturbances and preprocesses the chronological and nonchronological alarm messages that are used by the AAA, the BAI, and the BRS. The *CSI-R* handles restoration. It detects and corrects inconsistencies in the snapshot data file of the network, calculates the power flowing through the network, and makes this information available to the SRA and the UIA (see below).

This system design ensures that at least one agent performs all the tasks identified by the top-down analysis. The system achieves robustness by having multiple agents that can provide the same (or at least some) overlapping results. It obtains efficiency by the parallel activation of tasks. Reliability increases: even if one agent breaks down, the other agents can often produce a result which, although not as good as the one provided by the complete system, is still useful to the operator.

A cooperative scenario

We'll now present a scenario in which a disturbance occurs in the network.[5,9] Although the disturbance ultimately involves all the agents at some stage, we'll concentrate on

```
(ALARM-MESSAGES
   ((AGENT-ID "CSI")                         (DISTURBANCE-ID "UNKNOWN")
    (CHRONOLOGICAL "NO")                     (BLOCK-TYPE "MIXED")
    (BLOCK-ID "921211104820")
    (ALARMS
    (  "(INT 921211 104820 ERALTACH_ERA38IN CA ERA LOCAL)"
       "(INT 921211 104820 ERALACH_ERA38IC CA ERA LOCAL)"
       "(INT 921211 104820 SDPLACH_SDP38IN CA SDP LOCAL)"
       "(INT 921211 104820 STOLTACH_ERA38IN CA STO LOCAL)"
       "(INT 921211 104820 ACHLTACH_ERA38IN CA ACH LOCAL)"
       "(PROT-LIN 921211 104820 ERALTACH_ERA38RTIE ERA LOCAL)"
       "(PROT-LIN 921211 104820 STOLTACH_ERA38RTIE STO LOCAL)"
       "(PROT-LIN 921211 104820 SDPLACH_SDP38RTEM SDP LOCAL)"
       "(PROT-LIN 921211 104820 ACHLTACH_ERA38RTEM ACH LOCAL)"
       "(OSCILOS 3)"))))
```
(a)

```
 (ALARM-MESSAGES
   ((AGENT-ID "CSI")                         (DISTURBANCE-ID "UNKNOWN")
    (CHRONOLOGICAL "NO")                     (BLOCK-TYPE "MIXED")
    (BLOCK-ID "921211104820")
    (ALARMS
      ("(INT 921211 104820 ERALTACH_ERA38IN CA ERA LOCAL)"
       "(INT 921211 104820 ERALACH_ERA38IC CA ERA LOCAL)"
       "(INT 921211 104820 SDPLACH_SDP38IN CA SDP LOCAL)"
       "(INT 921211 104820 STOLTACH_ERA38IN CA STO LOCAL)"
       "(INT 921211 104820 ACHLTACH_ERA38IN CA ACH LOCAL)"))))
```
(b)

```
SM AAA                                        ;; Portion of AAA's self model
   (NewAlarms
    UpdateTopologyWithSnapshots
    DiagnoseFaults)                           ;; list of skills
   (SNAPSHOT
    DISTURBANCE-ID
    ALARM-MESSAGES
    GENERATED-HYPOTHESES
    VALIDATED-HYPOTHESES
    INITIAL-AREA-OUT-OF-SERVICE)              ;; information of interest to the AAA
```
(c)

Figure 2. Handling alarms: (a) the format for the alarm messages arriving at the alarms-analysis agent; (b) alarms related to the change of state of the breakers; (c) alarms-analysis agent's self model.

the diagnosis work of the AAA and the BAI. We'll somewhat cursorily cover the roles of the CSI (which behaves as the source of information for the rest), the BRS (which assists with the diagnosis), and the SRA (which devises a restoration plan).

The scenario starts from the point at which the CSI sends alarms to the AAA and the BAI. Figure 6 in Part 1, p. 69, illustrates the interest descriptor for the CSI's acquaintance model, which shows that the AAA should receive all nonchronological alarms and the BAI should receive alarms that relate to the change of state of the breakers (that is, alarms that start with "**INT**").

Figure 2a shows the format for the alarm messages arriving at the AAA (for the definition, see Part 1, p. 69, Figure 7). At the same time, the alarms related to the change of state of the breakers arrive at the BAI (see Figure 2b). When the AAA receives the alarms, its planning and coordination module (PCM–see Part 1, p. 68) examines its self model to determine what, if anything, to do with the unrequested information (see Figure 2c).

The PCM sees that **ALARM-MESSAGES** are of interest and checks exactly what they can be used for. One of the checks performed on all interesting information that is received is whether it is a trigger for a skill. In this case, **ALARM-MESSAGES** is indeed a trigger (see Figure 3) for the **NewAlarms** skill (**DISTURBANCE-ID** is also a trigger for this skill). The associated conditions are

- that the CSI agent sends **ALARM-MESSAGES**, that they contain the block type **UNIQUE**, and that the CSI has also sent an accompanying **DISTURBANCE-ID** (this condition is not true);
- OR that the received **ALARM-MESSAGES** contain the block type **MIXED** (this is true).

Because the final clause of the OR condition is met, the AAA triggers its **NewAlarms** skill (see Figure 3). While this is going on, the BAI performs a similar pattern

```
TRIGGER T_NewAlarms
  (ALARM-MESSAGES DISTURBANCE-ID)    ;; Inputs
  [IF (OR
        (AND (CONTAIN(ALARM-MESSAGES,"AGENT-ID "CSID""),
              CONTAIN(ALARM-MESSAGES,"BLOCK-TYPE "UNIQUE""),
              CONTAIN(DISTURBANCE-ID,"AGENT-ID "CSID""))),
        CONTAIN(ALARM-MESSAGES,"BLOCK-TYPE "MIXED""))
     {EXEC(NewAlarms);}]
```

Figure 3. Trigger specification for the **NewAlarms** skill.

```
SM BAI                        ;; Portion of BAI's self model
   (UpdateTopologyWithSnapshot
    UpdateBAITopology
    InitialBlackOutArea)   ;; list of skills
   (ALARM-MESSAGES
    DISTURBANCE-ID
    SNAPSHOT)                 ;; information of interest to the BAI
```

Figure 4. The BAI agent's self model.

of reasoning based on the set of alarm messages that it has received. Again, by looking at its self model, it realizes that it is interested in the unrequested information that has just arrived (see Figure 4).

Looking at its behavior-triggering conditions, the BAI realizes that the condition related to the information just received is that of the **InitialBlackOutArea** skill (see Part 1, p. 67 for a definition of behavior and skill). In this case, however, the trigger cannot activate until both inputs are available—expressed by the "**PRE**" keyword, which stands for precondition. Consequently, the trigger suspends, awaiting the corresponding **DISTURBANCE-ID** (see Figure 5).

As a consequence of the satisfaction of the AAA's **T_NewAlarms** trigger, the AAA's PCM instructs its monitor (see Part 1, p. 66 for a definition of the monitor) to start executing the **NewAlarms** skill (see Figure 6). In this case, the mandatory input (**ALARM-MESSAGES**) is available, but none of the optional inputs are. The behavior's body executes first—it has two alternative branches. The first branch checks whether the **ALARM-MESSAGES** have the block type **MIXED**. They do, so the monitor executes a **seize** command that instantiates a semaphore (called **EXEC-BLOCK-ANALYSIS**) to ensure that no more alarm messages will be analyzed until the complete analysis of the current messages finishes.

If this condition is not satisfied—which is not the case in this scenario—then the **mSetAlarmMessages** monitor unit (MU—see Part 1, p. 66) executes—this introduces the alarms to the intelligent system (IS—see Part 1, p. 65). The analysis of any new alarms is then blocked as before, and the monitor waits five seconds (to see whether the optional input of **DISTURBANCE-ID** is forthcoming from the CSI) before continuing this behavior's execution.

In either case, when the behavior's body is complete, the child behaviors activate. The first to activate is **OngoingFault** (see Part 1, p. 67, Figure 4), which checks whether the alarms belong to a previous fault. In this case, the alarms have **UNKNOWN** as a disturbance identifier; hence, they do not correspond to an existing fault, so this branch fails. Similarly, the **Maneuvers** child fails because **DISTURBANCE-ID** does not equal **MANEUVERS**. This means the third child, **StartNewDiagnosis**, starts. (**StartNewDiagnosis** has the same structure as the **DiagnoseFaults** skill—see Part 1, p. 68, Figure 5). Both representations (**StartNewDiagnosis** and **DiagnoseFaults**) are required because the AAA's fault-diagnosis process can be started directly by its PCM without first checking **OngoingFault** and **Maneuvers**).

The **StartNewDiagnosis** behavior first executes the **SetNewFault** MU (see Part 1, p. 66, Figure 3), which produces this **DISTURBANCE-ID** as an intermediate result (see Figure 7).

The monitor returns all intermediate results to the PCM, which evaluates whether they can be used elsewhere in the agent or even disseminated to acquaintances. To enact dissemination, the AAA's PCM examines its acquaintance models to determine whether any other agents are interested in receiving the freshly generated information. In this

```
TRIGGER T_InitialBlackOutArea
    (DISTURBANCE-ID ALARM-MESSAGES)
    [IF (AND
              PRE(DISTURBANCE-ID),
              NOT(CONTAIN (DISTURBANCE-ID, "DISTURBANCE-ID \"MANEUVERS\"")),
              PRE(ALARM-MESSAGES))
       {EXEC(InitialBlackOutArea);}]
```

Figure 5. Trigger specification for the InitialBlackOutArea skill.

```
BHVR NewAlarms
   (ALARM-MESSAGES)                                              ;; Mandatory inputs
   ( DISTURBANCE-ID
     GENERATED-HYPOTHESES
     INITIAL-AREA-OUT-OF-SERVICE
     VALIDATED-HYPOTHESES)                                       ;; Optional inputs
   (VALIDATED-HYPOTHESES)                                        ;; Results
   (OngoingFault                                                 ;; Child behaviors
    Maneuvers
    StartNewDiagnosis)
   ( (END "contain.(ALARM-MESSAGES.\"BLOCK-TYPE \\\"MIXED\\\"\") "  ;;Condition 1
          "seize.EXEC-BLOCK-ANALYSIS")
     (mSetAlarmMessages "seize.EXEC-BLOCK-ANALYSIS"              ;; Condition 2
                        "wait.5" ((END "" ))))
```

Figure 6. Specification of the NewAlarms skill.

```
(DISTURBANCE-ID
  ((AGENT-ID "AAA")
   (DISTURBANCE-ID "921211104820")
   (BLOCK-ID "921211104820")))
```

Figure 7. `DISTURBANCE-ID` intermediate result.

```
AAM BAI                         ;; Information of interest to the
  (ALARM-MESSAGES               ;; BAI as perceived by the AAA
   DISTURBANCE-ID)
```

Figure 8. A portion of the AAA's model of the BAI.

case, the PCM sees that the BAI is interested in receiving **DISTURBANCE-ID**s in all circumstances, so the PCM duly sends it as unrequested information (see Figure 8).

When the BAI receives this **DISTURBANCE-ID**, it refers to its self model and determines that it is interested in receiving **DISTURBANCE-ID**. So, it checks whether **DISTURBANCE-ID** will trigger any behaviors. This information allows the partially activated **T_InitialBlackOutArea** trigger to complete (because **PRE(DISTURBANCE-ID)** is now true and **DISTURBANCE-ID** does not contain **MANEUVERS**). This means that the monitor invokes the **InitialBlackOutArea** behavior (see Figure 9).

This behavior's body consists of three MUs that activate sequentially. The first, **mReadNonChronologicalAlarmMessages**, sends the received alarm messages to the IS, and locks (using the **EXECBAI-2** semaphore) the analysis of any new alarm messages until the analysis of the current ones finishes. The **mInitialAreaOutOfService** MU then activates, which invokes the IS task that calculates the initial area out of service. After this finishes, the **mClearBAI** MU activates, which executes an IS task that erases the alarm messages and all the information derived from them (providing a fresh start for the analysis of new alarm messages). The monitor returns the behavior's results in the structure **INITIAL-AREA-OUT-OF-SERVICE**, which provides an ordered list of the elements that are likely to be in the area out of service. Each element in the list receives a certainty coefficient because the use of misleading information such as signaling failures or triggering errors causes uncertainty about which elements are at fault. In this case, the result states that the initial area out of service consists, in decreasing order of certainty, of the elements **LTACH_ERA38**, **LACH_SDP38**, **ACH38A**, and **ACH38B** (see Figure 10).

The monitor then passes this result up as the outcome of the **InitialBlackOutArea** behavior. The BAI's PCM checks its acquaintance models to see if other agents are interested in the information. The PCM finds out that **INITIAL-AREA-OUT-OF-SERVICE** is of interest to the AAA in all circumstances, so the PCM sends it as unrequested information (see Figure 11).

Meanwhile, the AAA has continued its **StartNewDiagnosis** behavior, updating its topology description with the latest batch of alarms, and then performing hypothesis generation. (Generation is from scratch or is based on the **GENERATED-HYPOTHESES** supplied by the BRS agent). The **GENERATED-HYPOTHESIS** structure contains an ordered list, based on the certainty factor associated with each hypothesis, of the elements likely to be at fault (see Figure 12).

When hypothesis generation completes, the **StartNewDiagnosis** behavior has finished. However, because the behavior has two child behaviors (**RefineHypotheses** and **ValidateHypotheses**—see Part 1, p. 68, Figure 5), the monitor considers these next. **RefineHypotheses** is considered first. Assuming the unrequested information pertaining to the **INITIAL-AREA-OUT-OF-SERVICE** has arrived from the BAI, the AAA's PCM inserts it into the local context of the **StartNewDiagnosis** behavior (and all the children) because this information is defined as one of the behavior's optional inputs. In this case, the leftmost branch of the **Refine-**

```
BHVR InitialBlackOutArea
  (DISTURBANCE-ID ALARM-MESSAGES)                              ;; mandatory input
  ()                                                           ;; optional input
  (INITIAL-AREA-OUT-OF-SERVICE)                                ;; results
  ()                                                           ;; child behaviors
  ((mReadNonChronologicalAlarmMessages "seize.EXECBAI-2" (     ;; Body
    (mInitialAreaOutOfService "" (
      (mClearBAI "" (
        (END "release.EXECBAI-2" ))))))) )
```

Figure 9. Specification of the `InitialBlackOutArea` behavior.

```
(INITIAL-AREA-OUT-OF-SERVICE
  ((AGENT-ID "BAI")
  (DISTURBANCE-ID "921211104820")
  (HYPOTHESES ( (~HYPOTHESIS_T ((ELEMENT "LTACH_ERA38") (CF 90.0)))
                (~HYPOTHESIS_T ((ELEMENT "LACH_SDP38") (CF 50.0)))
                (~HYPOTHESIS_T ((ELEMENT "ACH38A") (CF 30.0)))
                (~HYPOTHESIS_T ((ELEMENT "ACH38B") (CF 15.0))))))
```

Figure 10. `Initial-Area-Out-of-Service` result.

`Hypos` plan (see Part 1, p. 66 for a definition of plan) activates because `INITIAL-AREA-OUT-OF-SERVICE` is available (if this information was unavailable, `Refine-Hypotheses` would fail because the monitor has received no validated or generated hypotheses from the BRS). This in turn executes the `REF1` MU, which produces a list of `REFINED-HYPOTHESES`. This MU uses the information supplied by the BAI to reorder the list of (generated) hypotheses so that the fault's most likely cause is at the list's beginning (see Figure 13).

After `RefineHypotheses` successfully completes, its child behavior, `ValidateHypotheses`, executes. The `Diagnose` plan takes each of the (refined) hypotheses in turn and invokes sophisticated reasoning to determine whether it is the faulty element. In this scenario, the plan's execution substantially speeds up because the order in which it processes the hypotheses reflects the BAI's know-how, over and above that of the AAA's own hypothesis-generation activity (see Figure 14).

In addition to speeding up hypothesis generation (see Part 1, p. 66, Figure 3), the BRS can profitably interact with the AAA when they have both produced their respective validated hypotheses. In this case, the AAA considers the BRS' validated hypothesis before informing the operator about the results of its diagnosis. If both agents agree on the same hypothesis, the confidence in this result can be increased. However, if they disagree, a new phase of interaction must start to resolve the difference in opinion.

The restoration process activates whenever a disturbance is detected. Once the disturbance is identified, the CSI-D sends the disturbance identifier to the CSI-R. The CSI-R acquires the snapshot of the network, corrects any inconsistencies that have arisen in its representation, and calculates the current state's power-flow solution. The CSI-R then passes this information to the SRA so that the SRA can prepare for its restoration planning. The SRA waits until the diagnosis agents have informed it of the element suspected as being at fault (`VALIDATED-HYPOTHESES`) and then prepares a restoration plan. If, during this plan preparation, the SRA learns that the faulty equipment is different from that originally indicated by either the AAA or the BRS, it replans the restoration, taking this information into account.

During diagnosis, the UIA presents users with both the tentative (early) list of suspected hypotheses and the final (validated) list. During restoration, the UIA supports a more participatory interaction between users and the agent community. The UIA presents the restoration plan to users. Users can then decide to modify the plan, run a detailed simulation to see how the plan affects the network's state, or ask for a new plan to be devised, taking into account new constraints that they specify. The UIA also supports a reporting functionality: control engineers can ask for the logs of the disturbances to be presented and analyzed.

THIS APPLICATION HAS BEEN IN operation in Iberdrola's North DCR since the beginning of 1994 and has afforded six primary benefits. First, the agent system gives better results than its stand-alone counterparts because it takes into account multiple types of knowledge and data and then integrates them consistently. Second, this ability to flexibly manage, at runtime, multiple sources of data and multiple problem-solving perspectives provides enormous robustness to the overall system. If one agent crashes, the others will still be able to provide some form of solution. Third, the system can provide some results more quickly because cooperation

```
AAM AAA                              ;; Information of interest to the
  (INITIAL-AREA-OUT-OF-SERVICE) ;; AAA as perceived by the BAI
```

Figure 11. A portion of the BAI's model of the AAA.

```
(GENERATED-HYPOTHESES
   ((AGENT-ID "AAA")
    (DISTURBANCE-ID "921211104820")
    (HYPOTHESES
              ((~HYPOTHESIS_T ((ELEMENT "LACH_SDP38") (CF 50.0) (ORDER 1)))
              (~HYPOTHESIS_T ((ELEMENT "LTACH_ERA38") (CF 40.0) (ORDER 2)))
              (~HYPOTHESIS_T ((ELEMENT "ACH38A") (CF 35.0) (ORDER 3)))
         ......))))
```

Figure 12. `GENERATED-HYPTHOSESES` result.

```
(REFINED-HYPOTHESES
   ((AGENT-ID "AAA")
    (DISTURBANCE-ID "921211104820")
    (HYPOTHESES
              ((~HYPOTHESIS_T ((ELEMENT "LTACH_ERA38") (CF 80.0) (ORDER 1)))
              (~HYPOTHESIS_T ((ELEMENT "LACH_SDP38") (CF 20.0) (ORDER 2)))
              (~HYPOTHESIS_T ((ELEMENT "ACH38A") (CF 15.0) (ORDER 3)))
        ...))))
```

Figure 13. `REFINED-HYPOTHESES` result.

provides a shortcut. Fourth, the functions of the different domain systems can be increased independently, which makes them easier to maintain (see, for example, the argument for developing the BAI). Fifth, control engineers have an integrated view of the results in which they are interested. Finally, because the system is open, new agents can be added incrementally.

```
(VALIDATED-HYPOTHESES
   ((AGENT-ID "AAA")
   (DISTURBANCE-ID "921211104820")
   (HYPOTHESES
        ((-HYPOTHESIS_T ((ELEMENT "LTACH_ERA38") (CF 94.0) (ORDER 1)
          (EXPLANATION ((-EXPLANATION_T
            ((BREAKER_FAILURES ())
            (PRIMARY_TRIPS ("ERALTACH_ERA38" "STOLTACH_ERA38"
                "ACHLTACH_ERA38"))
            (BACK_UP_TRIPS ())
            (ERRONEOUS_TRIPS ("SDPLACH_SDP38"))
            (UNJUSTIFIED TRIPS ())
            (ALARMS_RATE 100)))))))
```

Figure 14. **VALIDATED-HYPOTHESES** result.

One of the key features of this multiagent system is the way it handles fault diagnosis by using two different types of data (the nonchronological alarms used by the AAA and the chronological alarms used by the BRS) and two different points of view (the typical diagnosis approach of hypothesis generation and validation used by the AAA and BRS, and the BAI's monitoring approach, which provides a high-level view of the network's status). With this setup, it is possible to dynamically select the solution method that is best suited to the current situation. For example, if the BRS is operational but the AAA is not, the control engineer receives the solution created by the BRS. But if both the BRS and the AAA are running, the solution provided is the one on which they mutually agree.

Also, because multiple agents are trying to generate the same results, users can avoid repetition of certain tasks. For example, both the AAA and the BRS can provide **GENERATED-HYPOTHESES**. Consequently, if the hypotheses generated by the BRS are available to the AAA before it starts its own hypotheses generation, it does not have to execute this task; it can use instead the hypotheses from the BRS (see Part 1, p. 66, Figure 3).

Based on the experience of developing and installing this multiagent system, we foresee important application design improvements. The current system's first drawback is that the energy transport network covers a vast geographic area (there is a huge amount of topological information) and that it encompasses a number of different voltage levels. Because the network's behavior depends both on the voltage level and the geographic location, the main problem-solving agents (the AAA, BRS, BAI, and SRA) must contain and manage information about Iberdrola's entire transport network. So, the agents require a substantial amount of memory and computing resources because they search through such a large problem space. To combat this problem, in the system's next version the agents will work with smaller portions of the network. This modification will make the agents easier to debug and maintain, faster in execution, and more cost-effective in that they could run on PCs instead of workstations.

The second drawback is that all the agents have uniform knowledge of the network. For instance, the AAA applies virtually the same knowledge about protective relays to its 400-kV, 220-kV, and 132-kV levels. However, if each voltage level had one AAA (or BRS or SRA), customizing each level's domain knowledge would be possible. For example, agents could reflect potentially useful knowledge. For example, protective relays on the 400-kV level are more reliable than those at the 132-kV level, and the 400 kV network is more interconnected than the 132 kV network and has more complex breaker structures (such as central breakers or rings of breakers). The system currently masks a further source of heterogeneity. The network itself is the result of the fusion of a number of smaller transport networks that were developed by different companies before coming under Iberdrola's umbrella. For example, the protective relays of the northwestern Iberdrola network are different from those in the rest of the network. We could exploit this information if we developed smaller and more specialized agents.

References

1. J. Barandiaran et al., "An Expert System for the Analysis of Disturbances in Electrical Networks," in *Operational Expert System Applications in Europe*, G.P. Zarri, ed., Pergamon Press, New York, 1991, pp. 145–166.
2. M. Emaldi et al., "Design and Operation of an Expert System Prototype for Fault Analysis in Electrical Nets," in *Perspectives in Artificial Intelligence*, Vol. 1, J.A. Campbell and J. Cuena, eds., Ellis Horwood, Chichester, U.K., 1989, pp. 145–152.
3. L.R. Larson et al., "The NSP Intelligent Alarm Processor Project," *Proc. Power Industry Computer Applications (PICA)-85*, IEEE Press, Piscataway, N.J., 1985, pp. 116–120.
4. K.S. Swarup and H.S. Chandrasekharaiah, "Model Based Fault Diagnosis of Power Systems," *Proc. ESAP (Expert Systems Applications to Power Systems) '91*, Univ. of Tokyo, Tokyo, 1991, pp. 223–227.
5. E. Abel et al., "A Multiagent Approach to Analyze Disturbances in Electrical Networks," *Proc. Fourth Int'l Conf. Expert Systems Applications to Power Systems*, CRL Publishing, London, 1993, pp. 606–611.
6. J. Barandiaran et al., "Diagnosing Faults in Electrical Networks," *Proc. EXPERSYS 91*, IITT Int'l, Gournay-Sur-Marne, France, 1991, pp. 63–68.
7. N.R. Jennings and T. Wittig, "ARCHON: Theory and Practice," in *Distributed Artificial Intelligence: Theory and Praxis*, N.M. Avouris and L. Gasser, eds., Kluwer Academic Press, Dordrecht, The Netherlands, 1992, pp. 179–195.
8. J. Corera et al., "On-Line Expert System for Service Restoration," *Proc. Fourth Int'l Conf. Expert Systems Applications to Power Systems*, CRL Publishing, 1993, pp. 119–124.
9. J. Corera et al., "A Cooperative Approach Towards the Solution of Complex Decision Problems in Energy Management and Electricity Networks," *Proc. Int'l Conf. Electricity Distribution*, Inst. Electrical Engineers, London, 1993, pp. 4.19.1–4.19.6.

The authors' biographies are on p. 70.