111

The JEVIS Service Platform — Distributed Energy Data Acquisition and Management

Peter Palensky
Vienna University of Technology

111.1 Introduction

The deregulation and liberalization of large parts of the European energy market had a number of essential implications. Besides the wanted and expected consequences like increased competition — and therefore decreased prices — the energy customers became more and more aware of energy as being a part of their daily financial balance. Having the opportunity to choose between different suppliers that offer in deed different "products" resulted in the need to know what and how much energy is consumed where and by whom.

The types of customer that is currently taking advantage of the liberalized energy market are typically small and medium enterprises (SMEs), multisite customers (e.g., supermarket chains), and medium industrial customers. Large customers and large industrial sites like metal or glass industry are — as a part of their customer retention efforts — looked after by the respective electric utilities' key account managers anyway, while the medium customers do not enjoy such attention.

These customers might change the supplying utility company but have no basis for this decision. It is generally not possible to estimate if tariff A is more suitable for some customer than tariff B unless you know the exact consumption behavior.

This demand for information was the starting point of the JEVis project (Java Envidatec Visualization). The Institute of Computer Technology (ICT) at the Vienna University of Technology (Austria) and Envidatec GmbH, a service provider in the domain of energy in Hamburg (Germany) created a concept of a new and modern data acquisition system that is capable of processing millions of new records per day and that offers high availability and scalability called the JEVis system (see also www.my-jevis.com).

0-8493-1985-4/05/$0.00+$1.50

Customers that join (i.e., that are connected to) this JEVis system can browse through and analyze the consumption of every process within their production site or shopping mall. This transparency enables them to optimize their business and to choose the optimal energy tariff. Some of the customers are subsequently equipped with an automatic load scheduling system, which helps to avoid expensive consumption peaks and can directly determine their savings via the JEVis system.

Modern energy bills for SMEs are based on the so-called load chart, a consumption chart with 96 samples per day (15-min consumption values). A typical tariff in Mid-Europe is that not only the consumed energy (counted by an energy counter, in kWh) but also the three largest 15-min consumption values (the consumption "peaks") are taken into account. Additionally, the power peaks constitute quite a large portion of the bill; thus, avoiding or decreasing these peaks can significantly reduce the energy bill. The JEVis system stores the energy consumption charts and offers the recorded data via a web-portal.

The next step after having a detailed load chart of the consumption is to find out what processes or what parts of the customer installation are responsible for a particular power peak or the load shape in general. As already mentioned, sometimes it is even possible by simply looking at the chart while knowing the customer's processes. Usually, it is, however, not that easy. Large buildings and industrial sites with a large number of electrical consumers sometimes cannot be overlooked in terms of power consumption. It is thus necessary to apply additional sensors, besides the existing energy meter, to increase the granularity of the information.

The JEVis system is, for instance, used for "benchmarking" (e.g., comparing) multiple branches of one business to identify the most efficient and the least efficient ones. This massively depends on how much information you "get out" of the building or the process and into your calculation and optimization algorithms. The sources of this information are a large number of sensors and actuators that are queried by a database that is capable of storing these large amounts of data. The main challenges in the JEVis project are the highly distributed nature of the customer installations. Long-range communication is entirely based on the Internet or other IP-based communication infrastructure like virtual private networks (VPNs). Topics like reliability and security are vital for the JEVis system, as well as how IP can be used for flat peer-to-peer networking as it is needed for "global" automation applications.

111.2 JEVis Architecture

The customer-side part of the JEVis system interconnects all necessary sensors and actuators at the customer's site via a local control network (also known as field area network or "fieldbus" [1]) that can be accessed via the Internet. See later in this chapter for a more detailed description of the respective communication infrastructure.

Having this infrastructure, the JEVis system can not only record and analyze energy consumption data but virtually any kind of measurement data. The usage of machinery, temperatures, the number of persons entering a supermarket, the status of the air conditioning equipment, and many other things are fed into the JEVis database for further processing. The customer can then not only see the load chart but also its correlation to other measurement charts and thus identify those parts of it facilities that are responsible for power peaks. The combination of process data and consumption data enables the customer to evaluate the performance of different branches of his enterprise and to optimize his overall business. The JEVis system offers an all-in-one solution to acquire, process, and use these data.

The IGUANA project (1999–2001, [2]) aimed toward generating the customer-side infrastructure for sensor networks and data acquisition. IGUANA is the direct predecessor to the JEVis project and resulted in a flexible Internet/fieldbus gateway that meanwhile runs on various different hardware platforms. The JEVis system uses the IGUANA gateway software on an embedded rail-mounted industrial PC called "VIDA." The key properties of this VIDA are

- Intel-architecture industrial PC
- Linux operating system with web-server, SSL (secure socket layer), SSH (secure shell), etc.

- Robust transaction-based nonvolatile flash memory, no moving parts
- Digital input channels for energy meters (electricity, gas, water, etc.)
- Digital outputs channels for relays
- LonWorks control network interface for additional sensors and actuators
- Ethernet or analog, ISDN (integrated services digital network) or GSM (global system for mobile communication) modem for IP-connectivity.

The VIDA hardware is designed for robustness and usage in a rough and EMC-critical (electromagnetic compatibility) environment. Therefore, all inputs and outputs (I/Os) are galvanically insulated. It has no moving or maintainable parts and can be installed in closed switchboards. When powered up, the IGUANA gateway software scans the attached LonWorks network for sensors and actuators. After the discovery procedure, each node that was found is registered in an on-board database and can thereafter be used and operated by the VIDA software.

Being originally intended to serve as a remote configuration node for the Envidatec Load Management System, it turned out that the customers were much more interested in an "auxiliary" feature of the VIDA: data logging. Meanwhile, providing historical measurement values and generating alarms depending on the system state and some rules is the majority of VIDA's usage.

The VIDA was initially used as an autonomous system, playing the role of an "agent" in terms of network management (= a server that simply responds to queries and simultaneously an alarm notificator that can initiate messages as well), using its on-board http-server and other server- and alarm-features. Now, the features and services of the VIDA are massively "grafted" and extended by including it in the JEVis system.

An alarm message is not only an E-Mail that is sent out by the VIDA anymore, it is part of a whole alarm management concept: th e JEVis system takes alarm messages from VIDAs and creates FAX messages, GSM SMS (short message service), telephone calls, and other ways to reach the intended recipient of the alarm message. If the receiving of the alarm is not confirmed within a defined time, further alarm messages are sent to other responsible persons until the problem is solved. Managing all this is too much for embedded nodes like the VIDA, but is no big deal for sophisticated platforms like the VIDA system.

These extended services were the driving factors for JEVis. The result is a flexible platform that can collect, calculate, distribute, and manage data that are typically acquired via some measurement system. The increased granularity of knowledge about a building or an industrial process makes it more easy to identify problems and to optimize the processes. Devices and facilities that were previously controlled and run independently can now be analyzed and viewed by using a global and correlating view. The shared energy supply always "links" the individual processes of a plant when it comes to energy consumption, although they are not controlled collectively. JEVis now enables the customer to obtain an overview and to identify and overcome logistical and organizational problems.

A typical situation that can be found at customer sites is sketched in Figure 111.1 A number of energy-consuming customer processes (manufacturing lanes, compressed air production, etc.) are controlled individually but supplied collectively. Since the behavior of the individual processes is not coordinated, unnecessary and unwanted consumption peaks may very likely occur.

The customer-side part of the JEVis system interconnects the individual devices and their controlling processes with a control network and schedules them with an energy management algorithm. Such energy management systems (EMS) coordinate processes in a more or less sophisticated way. Sometimes, like in the case of the Envidatec EMS, even a short-time prognosis of the individual consumers is used to optimize the consumption shape.

Interprocess coordination is, however, just one side of the medal. Even within the processes consumption, coordination is usually missing. Equipping the consuming appliances with the respective JEVis sensors that measure consumption and operation enables the customers to find out what parts of his system are responsible for consumption peaks. The detailed knowledge about the processes is then the basis for the configuration of the energy management.

Some of the JEVis customers are not consumers of electrical energy but rather producers. A typical example is a wind power station, which is usually not allowed to feed energy into the grid whenever possible. The respective policy of energy supply and production determines how much a particular wind power station might produce, depending on the time, the situation on the grid, and so forth. The optimization process here is no consumption optimization and load shaping but production and injection optimization. The algorithms are similar and the hardware equipment is almost identical. A control network measures the power production, the wind, etc. and feeds energy into the grid or not. Having logs about the production is essential, since produced energy is a valuable thing, in addition to the consumed energy.

The JEVis service provider side consists basically of a database that stores and processes the collected measurement data. The results are subsequently offered as Internet-content via web-interfaces to the customer. Hence, the JEVis system consists of three parts (Figure 111.2).

- The customer-side installation: the customer processes, equipped with an Internet-gateway (the VIDA, for instance) and probably some control network connected to this VIDA for additional sensors and actuators.
- The JEVis server, an Internet-able database.
- The customer as a client to the JEVis server, using a standard web-browser.

The VIDA gateway is, as already mentioned, an embedded rail-mounted personal computer, which is in charge of offering a remote control channel to a local control network, of evaluating alarm conditions, and of measuring and logging data like water consumption, temperatures, and so forth.

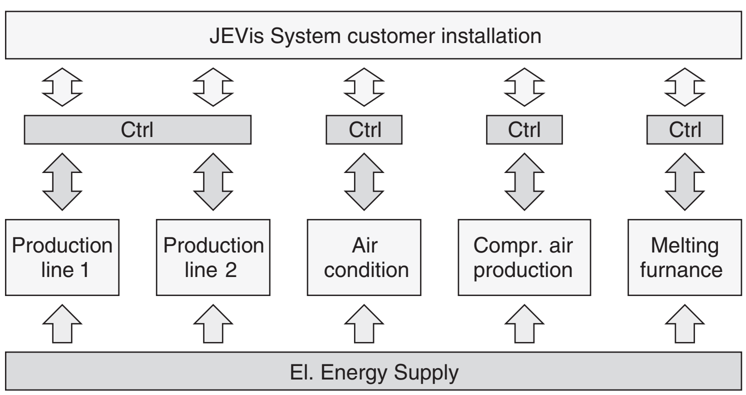

FIGURE 111.1 Separate processes linked via the JEVis system.

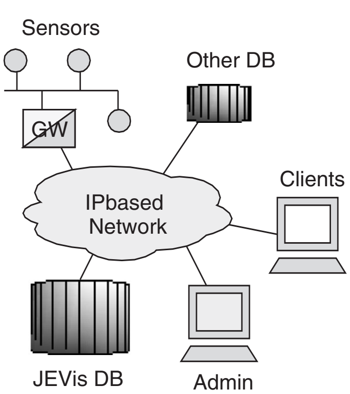

FIGURE 111.2 The JEVis database with its sinks and sources of data.

The main innovation of the VIDA is its network capabilities. Equipped with a flexible Linux-based set of software and some modem(s), it is able to

- pick up incoming calls and establish IP connectivity,
- to dial into a number of ISPs on-demand,
- to be permanently connected to an IP network like digital subscriber line (DSL), local area network (LAN), etc., and
- to be "woken up" by a telephone signal (e.g., a "ring") and to call back for IP connectivity.

These flexible networking options were the key to success. The VIDA can be installed virtually everywhere, in corporate networks, "in the field," etc. The customer's sites are subsequently equipped with VIDAs and the respective sensors. If only typical consumption data like electrical energy or heat consumption are needed, there is no need for a local control network, if temperatures, air pressure, and other values are needed as well; external sensor nodes must be attached. The protocols of the VIDA were carefully selected to safely pass firewalls and proxies while offering strong data security. Since conventional proxies often only allow http (hypertext transfer protocol) traffic, the gateway protocols must be embedded in http. The security-relevant layers must therefore be either embedded in http as well or be based on widely accepted standard protocols like secure socket layer (SSL). The JEVis system does both: it has internal authentication and wraps the traffic in SSL, when public networks like the Internet are used for connectivity. Key distribution for the cryptographic parts of the security layers is based on electronic chip cards, which is beyond the scope of this chapter. See [3] for a discussion on security and fieldbus/Internet connectivity.

One of the main challenges of the system was availability and reliability. The domain of energy supply is dominated by the high quality of energy supply in the last decades. Electrical energy was always available and had a constant quality. This is what the customers in this domain expect from anything that has to do with energy. When it comes to electronics, communication systems, servers, and the Internet, we find a totally different level of quality. These systems are, due to their complexity, much more instable and endangered by hardware and software problems than the proved components of energy supply. Therefore, all software parts of the JEVis system are monitored by watchdog-like programs for proper functionality and valid states. Two communicating software components (a client and a server), for instance, mutually monitor each other for timeouts, stability, correct behavior, valid status, and other things. Only these measures made it possible to offer the required availability. Figure 111.3 depicts two replicated instances of the JEVis system that exchange data for replication purposes and additionally monitor the system behavior of the peer besides two processes inside a system that have a client/server relation.

The monitoring occurs in a variety of ways. Typically, customer queries are issued (emulated) and the result is analyzed. This guarantees that at least the customer requests are processed correctly. Additionally, there are system-internal channels for supervision, which provide a deeper insight into the current state of the observed system. Key aspects like memory usage, system load, and other operating system-level attributes can be transported via these channels as well as state information of the individual processes of the systems. Figure 111.3 shows two processes on one and the same or even (or say "even better") on individual machines of the JEVis network. The processes are usually part of a client/server relation. Operational data are transmitted via these relations as well as monitoring data. In this way, one process A can be a watchdog for another process B. Since all parts of the JEVis system use or can use secured interprocess communication channels, the individual processes might be located on geographically

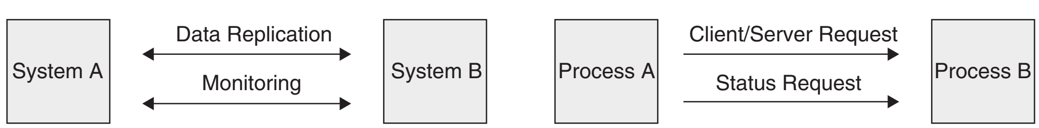

FIGURE 111.3 Systems and processes monitor each other.

distributed servers. This fundamental design decision makes it possible to install the system in a distributed and fault-tolerant way. One important rule of mutual supervision is that the final decision chain of "who resets who" (in the case of a detected faulty state) must be strictly hierarchical in order to prevent loops and meshes in the decision process. The propagation of rollback or reset actions must be absolutely deterministic. The internal client/server relations for the status requests and reset commands of all components must therefore follow a strict hierarchy.

111.3 Distributed Data Acquisition, Storage, and Access

Data are acquired via a number of different channels since the gateways can establish outgoing connections as well as incoming ones as already mentioned. If, for instance, one ISP is not reachable for a gateway, it tries to contact other ones and probably even via another channel (a GSM modem as a backup solution for the ordinary ISDN connection).

Let us take the incoming modem connection ("pick up" by the gateway) as an example. A connectivity component of the JEVis database calls the gateways via modems and transfers data via the subsequently established IP point-to-point connection. The JEVis network in Figure 111.4 has two replicated instances of its database and all its components (JEVis System 1 and JEVis System 2).

It serves two customer sites (customer A and customer B); one of the services might be to read out the data loggers of the gateways once a day. Let us suppose that the customer's sites are in different countries (customer A and JEVis System 1 in the same country as well as customer B and JEVis System 2) and the connectivity costs money, depending on the distance (public telephone network). The JEVis system therefore chooses the instance with the lowest on-line costs for each particular data retrieval job so that customer A will be queried by JEVis system 1 while customer B is read out by JEVis system 2.

The databases in the JEVis system instances synchronize with and replicate each other — some parts permanently, some parts at defined times or system states. The problems arise when parts of the JEVis instances are considered to fail. If one session that retrieves some thousands of data samples breaks because of hardware reasons or something similar, another data retrieval component has to finish the job. Thus, it might happen that JEVis system 2 continues a job that was initially started by JEVis system 1. System 2 therefore checks the status of the retrieval queue of system 1 and vice versa. If a job hangs or fails, the partner system tries to continue it. A transaction-oriented retrieval, storage, and replication policy ensures that the databases remain consistent.

The database has a number of storages and registries that are beyond the scope of this chapter, but two important data storages shall be explained as examples. The first one is the "samples storage." Databases for SCADA (supervisory control and data acquisition) like the JEVis system have to store "engineering values," that is, retrieved measurement values that are slightly more than plain numbers (See the ISO 16484 standard for an impression of the variety of datapoints that can be found in a modern building). A "sample" is an "extended" physical engineering value with a timestamp. The "extended" means that a single sample might

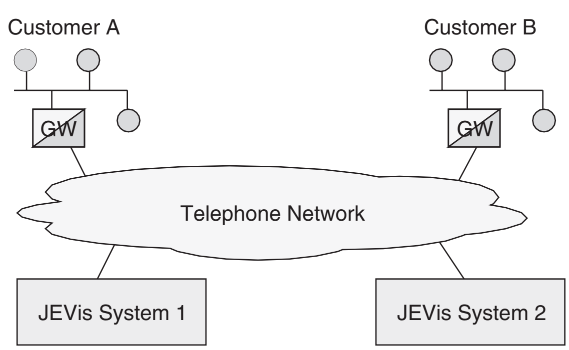

FIGURE 111.4 Geographically distributed customers and JEVis systems.

be a measurement value, a setpoint, a digital photo image, the status of a machine, a statistical, or organizational value typed in by some user and so on. Such samples typically consist of a set of attributes like

- relative accuracy of the value (in percent),
- absolute accuracy of the value (in the respective unit),
- physical unit in SI units,
- absolute accuracy of the timestamp (in sec),
- an absolute validity (from timestamp A to timestamp B),
- a relative validity: the interval in seconds around the timestamp (e.g., [0, 900[or]–450, 450]),
- the value itself, etc.

All JEVis services use these samples. Statistics are calculated, alarm conditions are evaluated, prognoses are estimated, etc., and all of this is stored in the samples storage. All entries in the samples storage provide additional information about the retrieval procedure so that it is possible to determine how a sample got into the sample storage. Samples are never deleted; they can only be updated by a newer sample with a more recent timestamp.

Data samples can be assigned to data rows like measurement rows (samples from one data source) or other rows like consumption samples from holidays or the temperatures of the 15th of March of the last 10 years.

Another important registry in the JEVis database is the "organizational hierarchy registry." Customers might see their sites and branches in different views. One example is that a supermarket chain sorts its branches by

- geographical location or
- financial clusters,

since a troubleshooting team for the freezers needs the first "view" of the sites, while the accounting department would use the latter one. Although both views finally contain the same sites, the views are different. Users that authenticate themselves to the JEVis portal can choose their preferred views and work with them.

The database replicates its tables and data structures depending on their contents and their role in the system. Configuration data like "data retrieval schedules," for instance, are replicated synchronously and immediately. Data storages are replicated on demand and asynchronously since they are unique in the system anyway. If two samples represent a value from the same data source, logged at the same time, a deterministic collision and conflict resolution algorithm cleans the samples storage.

Data are put into and read out of the database via components that tightly attach themselves to the database via standard interfaces like ODBC or JDBC (open database connectivity, Java database connectivity). These components typically reside on the same system and machine as the database itself. These components either have a client- or a server-interface to the rest of the system or to the outside. Data input components (DIs) and data output components (DOs) furthermore might be of an interactive nature or of a batch nature. The first one usually interfaces to humans, which results in the requirement of low latency and quick responses.

Figure 111.5 shows typical data input components and data output components. The HTTP Server DO, for instance, would be an interactive DO because it is typically queried by human users using a web browser. The SMTP Client DO is noninteractive since the simple mail transfer protocol (SMTP) does define any qualities of service regarding interactivity.

Server components are always queried by some client while client components might contact other servers. The server DIs and DOs are therefore the front-ends of the system, while the clients are back-ends that retrieve data or send data somewhere.

The simple object access protocol (SOAP) client DI in Figure 111.5 is a classical DI. It queries JEVis gateways for measurement data and other information from the customer site via SOAP. This client DI is used by the database and abstracts or encapsulates all gateway-relevant technology like special protocols, authentication mechanisms, datatype abstraction, and so forth. The HTTP Server DO in is, as already

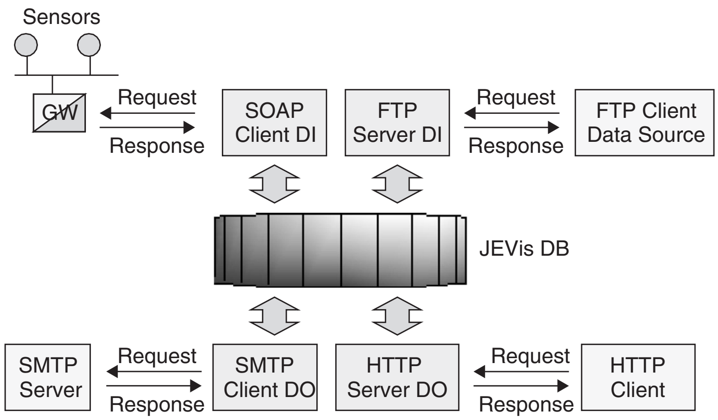

FIGURE 111.5 Paths of data into and out of the database.

mentioned, an interactive DO. Usability, response time, user interfaces (typically graphical user interfaces, GUIs), and other aspects are the main concert for developing these components, since they are the direct "face" to the customer that wants to be satisfied. Interactive DIs and DOs are therefore typically web-enabled forms and GUIs, based on Java servlet/applet technology.

The distributed nature of the system requires some considerations to protect data and other aspects of security. The data replication channels, for instance, should be protected from unauthorized access or manipulation. Therefore, JEVis implements the following methods:

- secure data transport between all communication entities and components of the system,
- secure data storage,
- powerful and flexible access right methods,
- usage of smart card security, and
- secure customer front-ends.

See [3] for further details on data security for control networks and smart card technology.

111.4 Global Energy Management

Considering influence on the consumption behavior is generally called energy management or more precisely demand side management (DSM). DSM usually consists of three parts, namely

- retrofitting (changing insulation, devices, etc.),
- logistical optimization, and
- load scheduling.

In our context, energy management means load scheduling, the active influence on the operation of equipment. The customer-side part of the JEVis system contains an energy management subsystem that actively schedules electrical consumers. The rules for this scheduling are based on the specific energy tariff. After having analyzed the customer processes with the JEVis system, and after having identified the reasons for power peaks and other unwanted behavior of the system, it is easy to configure and tune the energy management system in order to avoid this behavior.

Global energy management (GEM) goes one step further. The deregulated energy market in Europe moves toward a direction where customers are allowed to group their geographically distributed sites ("multi site customers") onto one energy bill. The overall consumption load is then used as a basis for the energy bill. The energy meters might record the individual load charts, which are then summed up, but the individual local energy management systems usually do not cooperate globally.

The problem of "global" energy management is basically the same as local energy management: a number of resource-consuming entities are to be organized and scheduled in order to impose some overall behavior. The difference to local energy management lies in the communication channels. The problems that a GEM application faces in the world of the Internet are perfectly transferable to other "global" automation problems. The weak quality of services, the limited ways of establishing network connectivity, and the restricted transport of information in the Internet are problems that any distributed automation application faces so that global energy management can be seen as an example.

Local EM interconnects the energy consumers or say the controllers of the energy consumers via some kind of local network — a control network or a LAN. The availability of these network resources is generally very high and not connected to any "on-line costs." The local EM peers can count on the fact that the connectivity is a permanent one, and every member of the EM system can reach any other member whenever needed.

GEM faces a different network infrastructure. Long-distance connectivity is costly; the nodes might be connected to the Internet via GSM channels, telephone lines, or other ways of "going on-line." The Internet is virtually the only affordable network that offers long-range peer-to-peer connectivity at reasonable prices, but it still costs. Therefore, the members of the GEM system will not stay on-line 24 h a day but only at certain times or on demand. Not only the frequency of data exchange but also the amount of exchanged data is relevant since it directly influences communication costs.

As a consequence, the JEVis GEM system estimates the future behavior of its members and uses this prognosis to calculate the schedule in an as-off-line-as-possible way. The algorithms of this optimization process will not be discussed here but another important aspect of the system: the need for flexible communication. The GEM system consists of a number of nodes that represent an energy-consuming or-producing entity like

- a wind power station,
- a private home with a fuel cell system,
- a bakery,
- an industrial site, or
- a supermarket with a number of freezers.

Some of these nodes might "store virtual energy" like the freezers or the bakery; others can produce energy like the fuel cell and the wind power stations, and the majority simply consume energy. Thus, the system consists of a geographically distributed number of energy consumers, storages, and producers with restricting rules of consumption, storage, and production. A wind power station, for instance, cannot be influenced in order to produce more energy and a bakery may not be switched off anytime but only under certain circumstances. Thus, the optimization process tries to take advantage of the degrees of freedom that the system offers in order to find an acceptable schedule for all members. Finding the right schedule that satisfies the "global goal" (some overall load chart, some special consumption peak avoidance in a region, etc.) strongly depends on the amount of information that the algorithm has about the situation. The degrees of freedom of every member sum up to a very complex optimization problem, which results in large amounts of data to be exchanged. Independent of the type of algorithm (negotiation-based, search for solution, etc.), the members actually need "flat" peer-to-peer connectivity in order to exchange data like tables, statistics, states, commands, etc.

The Internet usually offers peer-to-peer IP networking once the nodes are on-line. Unfortunately, depending on the Internet service provider, some of the dial-up nodes are hidden behind an IP proxy so that they cannot offer any services (open and "listening" IP) to the outside network. In this case, methods used by Internet Relay Chat (IRC) must be applied, where a publicly available server routes the requests of the GEM members that appear as IP clients. Figure 111.6, for instance, shows a wind power station and a gateway GW1 that want to communicate. Since the wind power station is behind a proxy, they use the JEVis relay to communicate instead of using native IP peer-to-peer communication. The same would occur if a GW goes on-line via standard GPRS (general-purpose radio switching, a packet-oriented GSM service), since a GPRS node gets an "internal" IP address behind a masquerading firewall

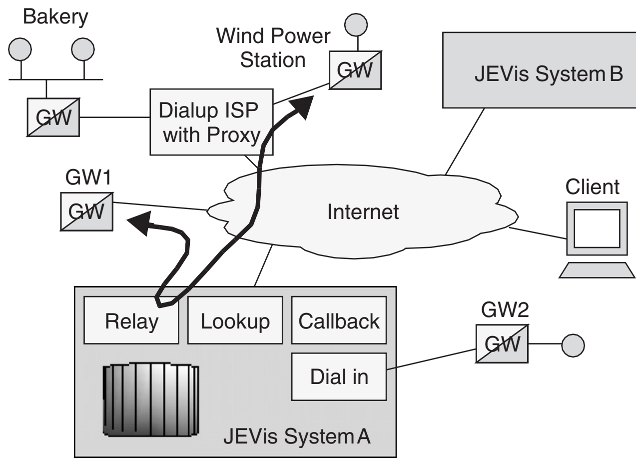

FIGURE 111.6 Connectivity services of the JEVis system.

that is not accessible from outside. Such firewalls, proxies, and other phenomena on the Internet restrict the communication massively and therefore the system must use technologies that are widely accepted and therefore supported and not inhibited by the Internet service providers. Additionally, the nodes might be organized in clusters and groups where not every node knows who and where its peers are. As a consequence, there must be some lookup tables where nodes can register in order to be found by other nodes.

The JEVis system therefore does not only host a database but also a number of other useful services that make a global automation application like global energy management possible:

- managing new dial-up connections (call back) on behalf of other nodes,
- dial-in service,
- lookup service for nodes to find peers, and
- communication relay for nodes with restricted transport (behind proxies, firewalls, etc.).

Another problem is the quality of services (QoS) like latency, availability, real-time aspects, and so forth. The IP protocol usually does not offer any guarantees. If the algorithm relies on the assumption that an offer (as a part of a negotiation procedure) is transmitted for sure within 5 sec, the algorithm is a bad choice for an Internet-based solution. Additionally, the Internet protocol itself is not the only unreliable part of the system. The lower-layer transports like dial-up telephone lines or GSM networks have a nondeterministic availability. It is not guaranteed that the node gets a free telephone line when it is needed. Therefore, the algorithm must be designed in such a way that it copes with unavailable information and unreliable and nondeterministic communication. The nodes must have as much "intelligence" as necessary to work fine even when communication is not possible for hours or days. This "local intelligence" results in a very robust and fault-tolerant system, which is one of the typical goals of distributed systems. The drawback of such a distributed architecture is that distributed network management is much more complicated. Therefore, the central JEVis system must be the location for network management-related things like node-configuration, diagnosis, and the like.

The distributed and robust nature of the GEM network would be destroyed if the JEVis system would not be redundant and distributed as well. If only one central server offers the communication relay and the other services, this server represents a single point of failure. The replicated and distributed nature of the JEVis system is therefore a good contribution to the fault tolerance of the GEM system.

111.5 Outlook

The JEVis system is, although very mature and massively in use, still subject to intensive research and development. The current research and development activities are system availability, alternative database systems, and, as already described, the infrastructure for GEM, which is probably the biggest challenge.

The availability of the overall system is still more limited than necessary due to insufficiencies of the Internet technology itself. By using dynamic host resolving, automatic query relays, and other dynamic methods, we hope to further increase the fault tolerance and availability of the system.

The JEVis database is currently based on an Oracle 9i® system, and proved stability and high performance. The philosophy of the JEVis services, however, would actually suggest an object-oriented design and not a relational database. Object-oriented databases (OODBs) are expected to be more flexible when it comes to different types of data that have to be stored. As an example, the configuration of all equipment used in the JEVis system (IP gateways, sensors, fieldbus nodes, etc.) is stored in and maintained by the JEVis database. A new type of device can result in a change of the data structures because it demands new, specialized data fields to be stored. An OODB has more flexibility in that aspect and can abstract devices and encapsulate data structures.

Another interesting branch of database technology is active databases (ADBMS) [4]. Active databases and especially active real-time databases are preferably used for job shop scheduling, work flow management, and production processes. Reactive behavior, triggered actions, and the action scheduling features of ADBMS are actually ideal for the JEVis services — the JEVis system also reacts to various changes in its environment. However, Oracle triggers are currently sufficient for the reactivity of the JEVis system.

Extensions like new services for customers, new devices, and components like alternative gateways and protocols are further fields of development.

References

[1] Loy, Dietrich and Schweinzer, Eds., *Open Control Networks, LonWorks/EIA 709 Technology,* Kluwer Academic Publishers, Dordrecht, 2001. AQ1

[2] Lobachov, M. and P. Palensky, Bringing Energy-related Services to Reality, Proceedings of the International Conference on Energy Economics (IEWT01), Vienna, Austria, 2001.

[3] Sauter, T. and P. Palensky, Security Considerations for FAN-Internet Connections, 3rd IEEE Workshop on Factory Communication Systems, Barcelona, 2000.

[4] Buchmann, A.P., Architecture of Active Database Systems, in *Active Rules in Database Systems*, Paton, Norman, Ed., Springer-Verlag, New York, 1998.