Netcentric Proxies for On-Orbit Sensors

Craig A. Lee, The Aerospace Corporation
Samuel D. Gasster, The Aerospace Corporation

This paper investigates the use of proxies to incorporate on-orbit sensors into netcentric environments. Proxies can provide a natural system interface that observes all of the tenets of netcentricity. Proxies can provide support for security, policy enforcement, reliability, mediation, power, performance, and operational management. Proxies can also support information assurance by providing a means to enforce the separation of system components based on security policy and practices. Proxies could even be used to determine the "personality" or "look and feel" of how on-orbit resources are exposed to external clients.

1 Introduction

With the growing influence of netcentricity, there is a desire in the space community to apply it to all system elements. Netcentricity entails information and services that can be discovered through a standardized messaging protocol and used by any person or system with the right authentication and authorization. Such netcentric operations are commonly supported through some type of SOA using the appropriate vocabularies, metadata schemas, and ontologies. When properly implemented, this approach can provide much better system extensibility and interoperability, and help avoid stove-piped systems and vendor lock-in.

However, not all system elements, such as on-orbit sensors, are amenable to direct exposure in a netcentric system. Since there is a fundamental trade-off between performance and flexibility, any system that must operate in a specialized, resource-constrained environment may not be able to fully support the requirements for netcentric communication and interaction. Furthermore, sets of on-orbit sensors may be part of a larger system that must be managed as a whole. For instance, any single on-orbit sensor may be on a vehicle with other sensors that interact and share local resources. Any single sensor could also reside on a module that is part of a fractionated cluster in which various modules are sharing resources [1]. Each vehicle could be in a constellation of vehicles that must be managed as a whole at some level.

Hence, in this paper, we will investigate the use of netcentric proxies to make on-orbit sensors available in a general service-oriented architecture, while transparently managing the constrained bandwidth, latency, orbital connectivity, and functional characteristics in an intelligent manner. That is to say, the netcentric proxy can actually expose the control and data of individual sensors to external users, but it can also expose an abstraction or higher-level interface to the sensor that is more appropriate and simpler for external users. We also note that netcentric proxies can also be used to virtualize on-orbit sensors, since users would not have to communicate with a specific hardware device at a fixed address, but could communicate through any instance of the appropriate proxy.

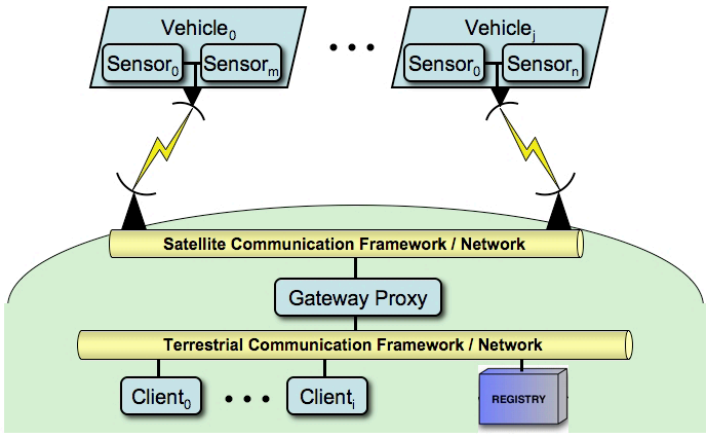

Figure 1. A Notional Satellite Gateway Proxy Architecture

2 Proxies for On-Orbit Sensors

As noted already, not all system elements are suitable to a netcentric SOA environment. On-orbit sensors operate in a highly constrained environment. On-board power is limited, communication is through highly specialized radio frequency (RF) and optical links, connectivity can be intermittent, and unique operation and usage policies are strictly enforced. On-orbit sensors, and their data, may also be classified at a higher level than other system components. The appropriate protections should be in place as data moves from the space segment, through the space-to-ground link, and into the ground segment.

For all of these reasons, directly incorporating on-orbit sensors in a netcentric SOA would be very problematic. SOAs typically require a common transport layer for communication (such as TCP/IP), services that "come and go" could cause disruption for clients, and an on-board sensor is probably not the place to enforce policy across competing requests from a potentially large number of clients. To deal with any of these issues on-board would require more on-board computing and power demand just to do "housekeeping".

It would be possible, however, to indirectly incorporate on-orbit sensors into a netcentric SOA by making them "available" through one or more gateway proxies. Figure 1 illustrates how such gateway proxies could be used to do this. Here a gateway proxy provides one or more services using a terrestrial SOA. This proxy is also connected to a satellite communication system that has the physical uplinks to vehicles and their sensors. These uplinks provide connectivity to multiple vehicles, each of which may have multiple sensors.

With this notional architecture, many issues and capabilities can be addressed, which we discuss in the following subsections.

2.1 Protocol Conversions

Proxies are a natural place to do protocol conversions between terrestrial networks and on-orbit vehicles since they are, by definition, in between the two. (This is called mediation in the parlance of netcentricity.) While work has been done in running common network protocols, such as TCP/IP, over delay tolerant networks, i.e., on an interplanetary scale, this will not be common. Existing systems will use unique, specialized communication and interaction protocols that are very different from those used in SOAs.

2.2 Addressability of Individual Sensors

In networks and SOAs, being able to address and send a message to specific recipients is a fundamental capability. Names and addresses of vehicles and sensors could be published to the registry that external systems are allowed to discover and use. Behind the gateway, however, these names and addresses could be mapped to whatever scheme makes the most sense internally.

2.3 Higher Level System Services

We also note that the externally visible names and addresses could, in fact, represent not just individual sensors, but also aggregate functionality provided by the sensors, vehicles, clusters, or an entire constellation. The use of gateway proxies would allow a range of services to be exposed on the terrestrial SOA—from individual sensors to higher level, aggregate services that define the apparent "behavior" or "personality" of the entire satellite system. As an example, a user may want infrared (IR) surveillance data with specific performance parameters. They could submit a request to an IR surveillance service that determines how best to satisfy this request with the available resources. The user could get an initial report describing how the request will be met, and if adequate, the user could resubmit the request for the actual data.

2.4 Managing Orbital Connectivity

Depending on the presence of cross-links in a particular satellite constellation, vehicles and their sensors may only have periodic connectivity to the ground. A gateway proxy could provide a continuous presence for the sensors, even if they are not over a ground station, thus providing a more robust client interface and experience.

2.5 Managing Reliability

Beyond just orbital connectivity, proxies could manage all aspects of the externally perceived reliability for sensors and vehicles. If the gateway can communicate with more than one ground station, it can reroute traffic if one ground station fails, or if the network link fails. In the event of a failure somewhere in the system the proxies could, at a minimum, provide information to clients about the failure. It can also attempt to transparently shield the client from failure by looking at alternate ways to satisfy service requests.

2.6 Power Demand

Some client service requests will require the expenditure of power onboard one or more satellites. The aggregate of client requests may, in fact, exceed the available on-board power. Hence, the gateway proxy would be where the best location to enforce energy policies could be enforced. By examining the client request stream, the proxy could rearrange or delay requests, when possible, to avoid excessive power demands.

2.7 Security and Information Assurance

Gateway proxies are also the natural "gatekeepers" for the on-orbit assets. They can fully participate in the SOA's security mechanisms and support information assurance. Clients must authenticate to the gateway to establish their identity and be authorized to request services and data from the satellites. Access is based on the client's role within mission operations and pre-defined usage policies. Proxies can also provide encryption, checksums, and other methods for monitoring data integrity.

2.8 Operational Policy Enforcement

Beyond issues of power management, security and information assurance, gateway proxies are also the place where all operational policies concerning on-orbit assets could be enforced. For example, proxies could enforce an operational policy of "do not slew the bore-sight of the sensor across the disk of the sun," or "do not exceed a given power/duty cycle," etc. Clearly proxies could be the policy enforcement point as part of an overall resource management and scheduling system.

2.9 Cluster and Constellation Configuration

Multiple sensors, modules, and vehicles may not be independent of each other and may have to be managed as a unified system. All sensors on a particular vehicle are related, since they must share common resources, e.g., power, communication bandwidth, etc. They may also be related through configurable functional attributes that are sensor-specific, e.g., band, filtering, etc. Hence, while clients may want to interact with individual sensors or with higher-level aggregate services, the proxy may have to manage the sensors as a group. (Essentially, this is enforcing configuration policy.)

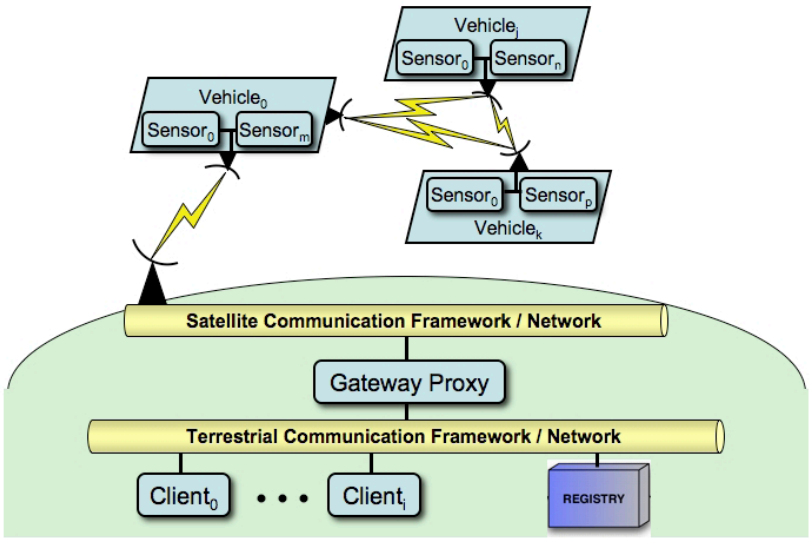

Figure 2. Gateway Proxy to Satellite Cluster Configuration

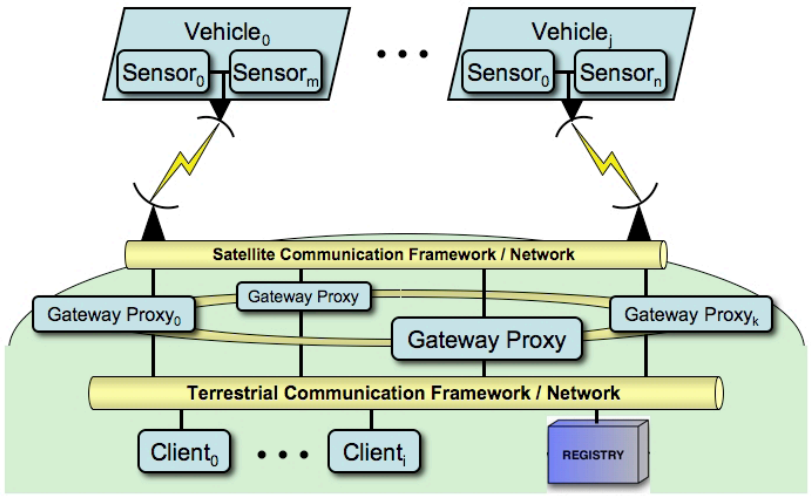

Figure 3. Satellite Gateway Proxies in a Peer-to-Peer Configuration

Likewise, sets of vehicles may have to be managed as a whole. Vehicles could be in a leader-follower configuration, a group or cluster of satellites, or in a multiple plane constellation. A specific example of satellite clusters is the DARPA System F6 program [2]. The idea behind F6 is to develop satellite architectures consisting of "future, flexible, fast, fractionated, free-flying spacecraft united by information exchange." This is illustrated in Figure 2, where a set of vehicles is in cluster flight configuration and communicating through their own RF cross-links. Each vehicle is a fractionated module with a specific set of functions provided for the cluster, i.e., the entire satellite system. When under attack, such modules can disperse and reform the cluster at a later time when it is safe to do so. If any one module fails, its functions could be taken over by another until a replacement module is available. Proxies would be very useful for interfacing such fractionated satellite architectures with a terrestrial service architecture.

2.10 Peer-To-Peer (P2P) Network of Gateway Proxies

In the discussion so far, we have presented the gateway proxy as if it were a single point of entry. The gateway could, in fact, have more than one "point of entry." There could be a P2P network of gateway proxies, as illustrated in Figure 3. Clients could contact the closest peer when requesting data or services from the on-orbit assets. (See "Terrestrial Data Archives.") The gateway peers could also provide redundancy and continuity of operations, i.e., reliability. The peers could be physically separated from one another such that if one peer crashes or is off-line for any reason, then access to the on-orbit assets is still possible by re-routing through another peer. One could also deploy a peer downrange on the battlefield to serve as the battlespace local point of contact.

2.11 Terrestrial Data Archives

Satellite sensors can produce tremendous amounts of data that must be served to clients and archived for future use. Such archives may be behind the gateway proxies or anywhere on the terrestrial SOA. If the archive is on the terrestrial SOA, the gateway could at least act as the agent that provides data to the archive.

If the archive is behind the gateway, however, then the gateway can serve as the gateway to the data archive as well. That is to say, the gateway could enforce data policy by managing access to the data, replicating data to different sites for faster access and reliability, and even providing data virtualization services. When a client requests satellite data from the gateway, it first looks to see if the requested data is available in the archive. If the requested data products are not available, then the gateway could actually schedule the on-orbit sensor to collect the raw data necessary to satisfy the client request.

2.12 Managing a Larger Sensor Network

Finally, we note that the on-orbit sensors could actually be part of a larger sensor network supplying data and information to a wide set of consumers. Consumers may want to interact with all of their data providers through a uniform model and interface to improve ease of use. On-orbit sensors may be only one of many data providers. Such an interface could define uniform ways for requesting data, specifying when the sensor produces data, and how the data is reported. One possible standard relevant to such sensor networks is the Sensor Web Enablement standard from the Open Geospatial Consortium [3].

3 Summary, Discussion, and Future Work

We have presented the concept of using proxies to manage the exposure of on-orbit vehicles and sensors in netcentric sys-

tems. Building on established concepts in computer networks and distributed systems, we argue that proxies on SOA—as an intermediary between clients and on-orbit assets—provide a mechanism to implement a wide range of important and useful capabilities. These capabilities include information assurance, policy enforcement, reliability, mediation, power, performance, and operational management. This can also be extended to managing how the "personality" or "look and feel" of vehicles and sensors are presented to external clients.

The concept of netcentric proxies for on-orbit vehicles and sensors has significant value, but clearly more thorough studies should be done to evaluate the possible difficulties of implementation and the actual benefits. For any specific systems, the general issue of increased latency introduced by a proxy would have to be evaluated. Also, any implementation in a real-world satellite system would carry with it any number of conflicting goals and design compromises. These conflicting goals and design compromises may have nothing to do with SOAs or proxies, but may impact their overall effectiveness.

To avoid pitfalls, it is clear that prototyping programs should be undertaken that start small and incrementally build capabilities for evaluation. The capabilities identified could be partitioned into phases that build on one another. Such prototypes could possibly leverage the Netcentric Core Enterprise Services [4] that are already being developed by the Defense Information Standards Agency (DISA). In addition to the engagement with DISA, the notion of netcentric proxies could also be promoted in defense contractor and community organizations, such as the Network-Centric Operations Industry Consortium [5], the Ground System Architectures Workshop [6], and the Federal SOA Community of Practice [7]. This would facilitate "closing the loop" among user/government requirements, standards organizations, and the vendor community.

Acknowledgments:

This work was supported by The Aerospace Corporation through the Innovation Grant Program of the Research and Program Development Office.

ABOUT THE AUTHORS

Dr. Craig A. Lee is a Senior Scientist at The Aerospace Corporation and current serving as President of the Open Grid Forum. Dr. Lee has worked in the area of parallel and distributed computing for the last 30 years. He has conducted DARPA and NSF sponsored research and served as a review panelist for the NSF, NASA, DOE, and INRIA. He has published over 60 technical works and sits on the editorial board of two journals. Dr. Lee holds a Ph.D. in Computer Science from the University of California, Irvine.

Computer Systems Research Department
The Aerospace Corporation, P.O. Box 92957
El Segundo, CA 90009
E-mail: lee@aero.org

Dr. Samuel Gasster is a Senior Scientist at The Aerospace Corporation, where he specializes in the application of high performance computing technology for scientific and remote-sensing applications, data-modeling and data-management system development, and systems and software engineering. He has worked at Aerospace for over 20 years and has supported a wide range of defense and civilian programs and agencies, including the USAF, NASA, NOAA, and DARPA. He currently supports the DARPA System F6 Program. His research interests include Quantum Information Science and Technology, new approaches to space mission systems engineering and complex systems. He has taught remote sensing and computer science courses at UCLA Extension and the Aerospace Institute. He holds a Ph.D. in physics from the University of California, Berkeley and an S.B. in mathematics from MIT.

Computer Systems Research Department
The Aerospace Corporation, P.O. Box 92957
El Segundo, CA 90009
E-mail: gasster@aero.org

REFERENCES

1. O. Brown and P. Eremenko. The value proposition for fractionated space architectures. AIAA Space 2006, (AIAA-2006-7506), 2006.
2. DARPA. System F6. <http://www.darpa.mil/TTO/ Programs/sf6.htm>.
3. The Open Geospatial Consortium. SensorWeb Enablement. < http://www.opengeospatial.org/projects/groups/sensorweb>.
4. Defense Information Systems Agency. Net-Centric Enterprise Services. < http://www.disa.mil/nces>.
5. The Network-Centric Operations Industry Consortium. < http://www.ncoic.org>.
6. Ground System Architectures Workshop. < http://sunset.usc.edu/GSAW>.
7. Federal SOA Community of Practice. < http://semanticommunity.wik.is/Federal_SOA_Community_of_Practice>.