# Wired Widgets: Agile Visualization for Space Situational Awareness

**Kelly Gerschefske**
*The MITRE Corporation*
**Jeremy Witmer**
*The MITRE Corporation*

## CONFERENCE PAPER

Continued advancement in sensors and analysis techniques have resulted in a wealth of Space Situational Awareness (SSA) data, made available via tools and Service Oriented Architectures (SOA) such as those in the Joint Space Operations Center Mission Systems (JMS) environment. Current visualization software cannot quickly adapt to rapidly changing missions and data, preventing operators and analysts from performing their jobs effectively. The value of this wealth of SSA data is not fully realized, as the operators' existing software is not built with the flexibility to consume new or changing sources of data or to rapidly customize their visualization as the mission evolves. While tools like the JMS user-defined operational picture (UDOP) have begun to fill this gap, this paper presents a further evolution leveraging Web 2.0 technologies for maximum agility. We demonstrate a flexible Web widget framework with inter-widget data sharing, publish-subscribe eventing, and an API providing the basis for consumption of new data sources and adaptable visualization. Wired Widgets offers cross-portal widgets along with a widget communication framework and development toolkit for rapid new widget development and data source integration, giving operators the ability to answer relevant questions as the mission evolves. Wired Widgets has been applied in a number of dynamic mission domains including disaster response, combat operations, and noncombatant evacuation scenarios. The variety of applications demonstrate that Wired Widgets provides a flexible, data driven solution for visualization in changing environments. In this paper, we show how, deployed in the Ozone Widget Framework portal environment, Wired Widgets can provide an agile, web-based visualization to support the SSA mission. Furthermore, we discuss how the tenets of agile visualization can generally be applied to the SSA problem space to provide operators flexibility, potentially informing future acquisition and system development.

## 1. INTRODUCTION

Our ability to collect data for Space Situational Awareness (SSA) continues to improve. New sensors and new sources, combined with service-based architectures, make more data available than ever before. With access to all of this data, it would seem that operators should have better comprehensive knowledge, understanding, and maintained awareness of space objects in orbits, of the space environment, and of existing threats/risks are concerned than at any previous time. However, situational awareness is not just access to an encyclopedia of information–it is the access to the right information, at the right time, when it supports operational needs. For the operator, the ability to adapt their tools to current needs is critical. Tools that support this kind of functionality and adaptation we refer to as a composition environment. As defined in [3], this is

> The overarching environment wherein users are empowered to compose new capabilities as needs arise. This environment spans organizations, space and time to provide the ability to share and reuse resources across organizations, from multiple on-going operations, from previous and steady-state operations.

## 2. THE NEED FOR FLEXIBLE VISUALIZATION

Efforts at Advanced Development Division (XR) within the 850th Electronic Systems Group (ELSG) realize that by leveraging a Service Oriented Architecture (SOA), their integrated Information Technology (IT)

infrastructure becomes agile and can rapidly respond to changing needs by employing loosely coupled and dynamic applications (i.e., services) [1]. XR provides a step in the right direction, but must be augmented with an agile user interface (UI) or visualization to leverage those new services as they come online. On current systems, the UI is a window into all the existing data sources that are known when that UI is developed. Fortunately for operators, but unfortunately for fixed UIs and systems, new sources of data come online regularly. Consequently, the UIs of today also need to be a window into all the data sources that are available after development is complete and the contract is closed. The SSA UI must be a composition environment as defined in previous section.

The key to a composition environment is the ability to compose and create new capabilities as needs arise. As the mission evolves, so must the SSA UI. By leveraging these new sources of data as they become available, the UI can better support the operator's needs. Other efforts have looked at the best way to represent and make data available for flexibility as seen in [5][1]. This research is focused on providing a UI that can be adapted to the user's needs, driven by the data, to provide situational awareness–one that is able to leverage those services and data, both known today and yet-to-be-developed as they come online.

Giving the operator a UI that they can change and adapt will empower the user to respond immediately to changes in the mission, requiring little or no external help from developers or support staff. If operators have the tools to discover and compose data together in a manner meeting their requirements, no additional effort must be expended to change or re-develop their tools. The closer the changes can be made to the operational edge, the less resources are required, those of manpower and cost. For example, an operator has an emerging need to visualize a set of breakup data from a new sensor both in time and space. In a non-flexible tool, this may require a requirements cycle, development, testing and certification, and deployment of a new tool. Instead, if the operator has the ability to mashup the breakup data with different visualizations in their toolset, such as a timeline or 3D globe, they can make the necessary changes to analyze the new data themselves.

## 3. WEB 2.0 AND WIDGETS

Web 2.0 technologies will play a central role in modernizing SSA visualization. Web 2.0 technologies are web development and design that facilitates interactive information sharing, interoperability, user-centered design and collaboration. In short, Web 2.0 technologies provide web users with the ability to create content and custom products while capitalizing on the power of the web to leverage collective intelligence.[5]

As evidenced by its rapid rise in the commercial world, Web 2.0 is no longer considered a new trend. It is a current reality, from web browsers to open-source tools, to the skills of developers[5]. Computer users at all levels are used to the ability to not just consume information on the web, but to interact, create, and communicate effectively through the web. From web-based communities, hosted services, web applications, to social networking, blogs, and wikis, the power to change and adapt is moving from developers to the end users. This cannot be done by building an application to a set of requirements[5].

We apply Web 2.0 to the SSA environment through Wired Widgets and its concept of a web widget. A web widget is generally defined to be a self-contained, portable, mostly client-side application that is designed to perform a specific function or set of closely-related functions. One example of a web widget might be a globe widget that is designed to consume a specific set of data formats, and display them on the globe for user visualization and manipulation. Widgets generally are designed to be reusable, hosted by any platform that is conforms to the correct specifications[6]. To provide this reusability, Wired Widgets runs across a number of portal environments, though the SSA implementation in this paper uses the Ozone Widget Framework (OWF) portal. For more details on OWF, which is a Government Open Source solution, see http://owfgoss.org/.

## 4. COMPOSABLE CAPABILITY ON DEMAND® RESEARCH

This research was funded by the MITRE Innovation Program, a MITRE internal research portfolio. Specifically, this research is part of the Composable Capability on Demand® (CCOD) portfolio of projects. There are two key facets of the CCOD® vision. The first is to empower the warfighter to leverage information as an effective weapon by providing the ability to rapidly combine, adapt and extend C2 capabilities in response to evolving threats and mission needs. The second is to introduce a new acquisition paradigm for IT-based capability in which we rapidly develop and field infrastructure, components and a method to employ them. Together, these elements offer a strategic advantage by enabling the user to innovate and define a more responsive, flexible, interoperable and robust C2 capability [5]. The research presented in this paper is focused on the former, to empower the operator with a flexible visualization tool allowing the user to dynamically combine, adapt, and display information in any domain. For more information on the broader CCOD portfolio, see [5] and http://www.mitre.org/work/areas/research.

## 5. WIRED WIDGETS

Wired Widgets is an Open Source[1], applications development toolkit and composable widget framework designed for the CCOD user composition environment. It provides operators with an agile means to leverage new and existing data, to perform data mashups and build custom, dynamic application displays.

In its simplicity, a Wired Widget is an OpenSocial Version 0.9 gadget augmented with the ability to intercommunicate and operate over dynamically generated data sets. It consists of a server side component for agile consumption of new data sources and a client side component that provide the flexible visualization described by this paper, which can be composed and combined together in various ways to meet the needs of operators. Fig. 1 shows the two components and demonstrates the interaction.

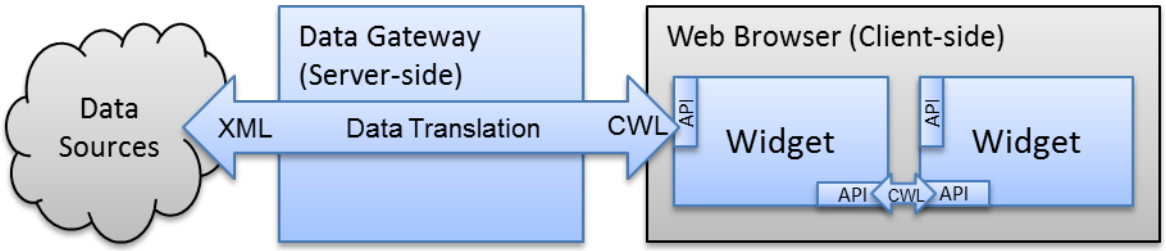


Fig. 1: Wired Widgets Architecture

The server component, referred to as the Data Gateway, provides an interface from the widgets to any and all data services. The Data Gateway interfaces with REST, SOAP, and other data services, including UDP streams. All data from the services is converted to a simple, flexible inter-widget format referred to as the Common Widget Language, or CWL. CWL is a minimalist approach to widget-based information exchange based on many of the proven benefits of existing data formats, such as Really Simple Syndication[2]. It is a hierarchical data format that allows for a common representation of both simple and complex structural information. The primary purpose of CWL is to serve as a common structure of data for the enabling of interoperable information exchange within a widget platform. Its core paradigm, very much like well adopted internet syndication technologies, is to provide a brief synopsis of information to one or more widgets along with a means for the widget to retrieve higher fidelity data within the language of a richer Community of Interest data model.

On the client-side, Wired Widgets includes a set of web widgets built on OpenSocial Version 0.9[3] gadget

[1] http://www.wiredwidgets.org, July 31, 2012
[2] http://www.rssboard.org/rss-specification, July 31, 2012
[3] https://sites.google.com/site/opensocialdraft/, July 31, 2012

specification. OpenSocial provides a cross-browser, cross-portal environment for building web widgets. As such, Wired Widgets run in a variety of portals, including, as demonstrated, the Ozone Widget Framework[4] portal. Each widget is a certain kind of visualization and can be combined in various ways to meet the needs of the operator or user. Wired Widgets provides an API for communication with the data gateway, inter-widget communication and common functionality, including setup and teardown of the widget functionality, and inter-widget eventing.

Fig. 2 shows the widgets running in the Ozone Widget Framework portal environment. The core widget, called the Data Manager, is shown on the left side of the screen. This widget interfaces with the Data Gateway through the Wired Widgets API, and "wires" the data out to the other visualization widgets, lending the framework its name. The CWL common format and inter-widget API allows for new widgets to be easily developed that operate in conjunction with the existing widgets.

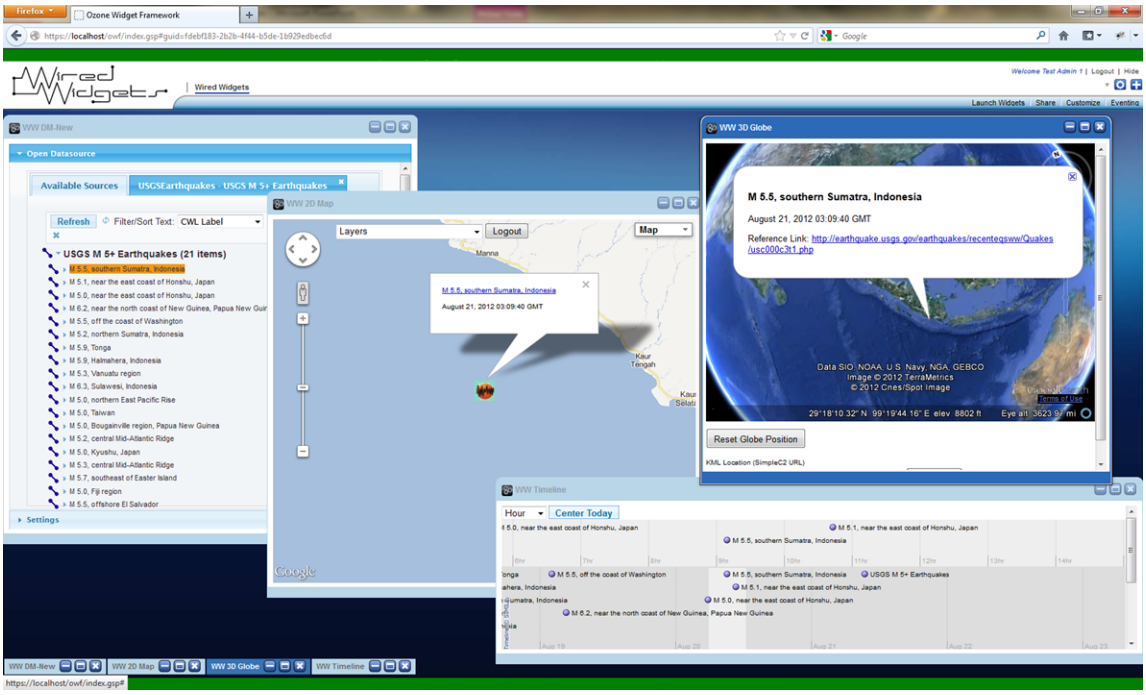


Fig. 2: Wired Widgets Running in OWF

## 6. AGILE VISUALIZATION WITH WIRED WIDGETS

Wired Widgets has provided agile visualization in a number of problem spaces, from Army C2 operations to unmanned aircraft control. Additionally, Wired Widgets showed effectiveness in Marine command post exercise [2].

Specific to the SSA space, Wired Widgets is applied to a JMS prototype capability that provides a set of net-centric services built on-top of a SOA infrastructure to expose satellite ELSETs, an ephemeris service, and conjunction analysis across a simulated catalog. For now the catalog is simulated, as research into high-performance conjunction analysis across a large catalog is still ongoing [4]. The Wired Widgets Data Gateway can interface with all three services, allowing the user to query pertinent information at any time. The Data Manager widget presents the services and service parameters to the user using the Wired Widgets API.

[4] http://owfgoss.org, July 31, 2012

Once a user selects to perform an all-on-all conjunction analysis against a satellite, for example, the Data Manager informs the Data Gateway of which service to call with a set of parameters. It then queries the JMS service, converts the result to CWL, and returns it to the Data Manager widget. The Data Manager widget then displays a list of objects that have some probability of conjunction. The user is then able to plot that result on other widgets, a 3D map or table as shown in Fig. 3.

Additional data sources were also exposed via service interfaces. The varied data standards are converted to the common CWL format by the Data Gateway, and then made available through the Data Manager to any other widgets that implement the Wired Widgets API.

Finally, the base set of widgets discussed at the beginning of this section did not provide the fidelity on a 3D globe to provide good SSA visualization. As part of the research, we rapidly developed a more fine-grained widget based on the NASA Worldwind toolkit [5]. Because the Wired Widgets API provides the data handling, data formatting, eventing, and widget management API, only the 3D visualization had to be developed. This dramatically decreased development effort. Additionally, new widgets are regularly developed against the Wired Widgets standard. As this library of widgets grows, it is more and more likely that a new visualization need can be fulfilled by an existing widget. To better facilitate this sharing and re-use, the Wired Widgets framework and a number of the associated widgets have been open-sourced under the Apache license. For more information, documentation, and code, see http://wiredwidgets.org.

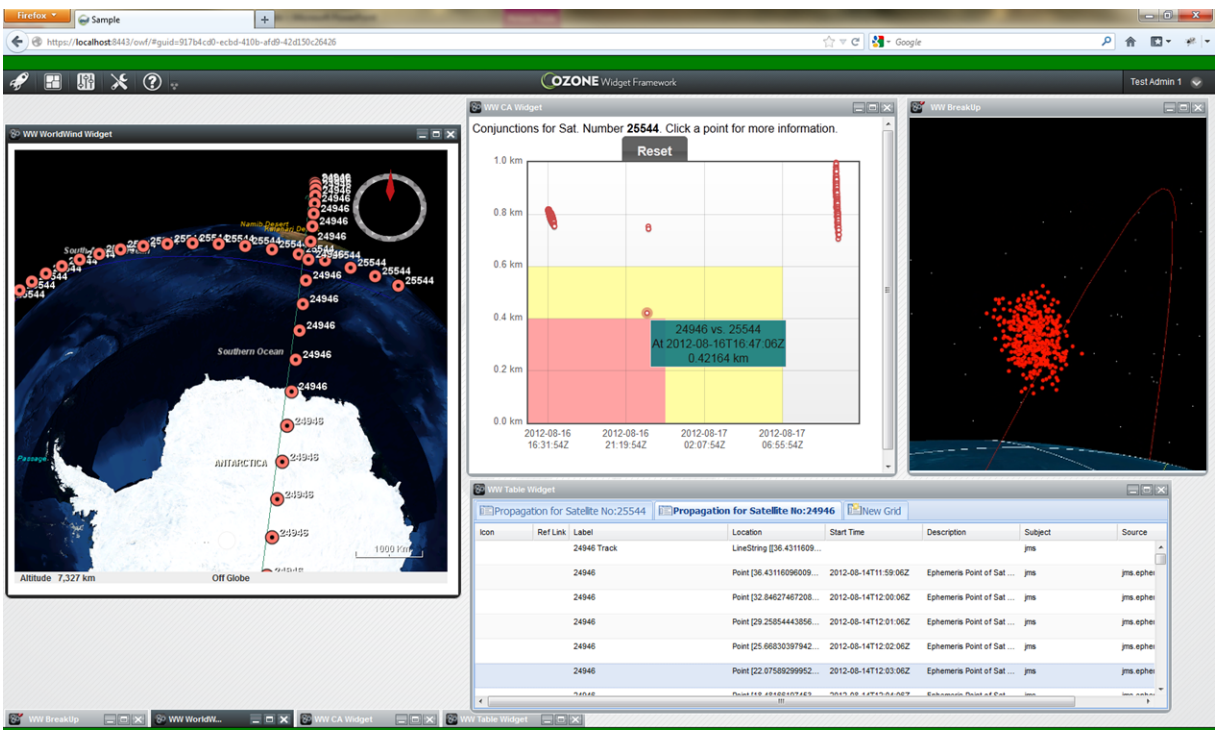


Fig. 3: Wired Widgets Applying JMS Services to Visualization Components

## 7. CONCLUSION

In this paper, we have identified a need for a agile visualization of SSA and introduced our strategies for the realization of a SSA composition environment. We have presented Wired Widgets, an applications development toolkit and composable widget framework, and addressed SSA visualization challenges by grounding their feasibility in a prototype we have implemented.

[5] http://worldwind.arc.nasa.gov/java/

We continue to demonstrate Wired Widgets implementations to the SSA community to encourage innovative thinking that will lead to changes in future acquisition and system development. In this way, SSA visualization applications can be modernized as effectively and efficiently as possible, resulting in increased operational capability.

Furthermore, we assert that Wired Widgets is simple enough to enable flexibility for a variety of current composition needs and flexible enough in design to enable changes in the API to accommodate progressively more complex uses. The goal is for the Open Source community to build upon the core framework if needed. For detailed information on how you can contribute to Wired Widgets, documentation and code access, see http://wiredwidgets.org.

## 8. REFERENCES


[1] Diane D. Buell, L. Shayn Hawthorne, and James C. Higgins. SSA Building Blocks – Transforming Your Data and Applications into Operational Capability. *Proceedings of The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2010.

[2] Hendrickson et. al. Operationalizing C2 Agility: Introducing Composable IT Environments to Command and Control. Technical Report MTR120995, The MITRE Corporation, May 2012.

[3] Robert Pancotti. Composable Capability on Demand (CCOD®) Terms of Reference. Technical Report MTR120223, The MITRE Corporation, May 2012.

[4] Mike Prausa and Norman Facas. An Application of Hadoop and Horizontal Scaling to Conjunction Assessment. Technical Report MTR 100411, The MITRE Corporation, 2010.

[5] J. D. Savage. Composable Capability on Demand (CCOD®): A New Paradigm for the Design, Acquisition and Employment of IT-Based C2. Technical report, ICCRTS, 2010.

[6] A. Soylu, F. Wild, F. Mödritscher, P. Desmet, S. Verlinde, and P. De Causmaecker. Mashups and Widget Orchestration. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 226–234. ACM, 2011.