

Loose Couplers as an Information Design Strategy

Dr. Robert W. Miller, Daniel G. Winkowski
The MITRE Corporation
Hampton, VA

Abstract

Net-Centric operations are predicated upon loose coupling among participants. Existing approaches to loose coupling have focused on the information and network interfaces, not on information design methodology. Cursor on Target has been described as a loose coupler that provides a strategy to exchange a minimal amount of structured, high impact information in a variety of use cases among battlefield participants. In this paper we provide an in-depth analysis of the loose coupler concept. We define a loose coupler as an information object design that optimizes data utilization (production and consumption) among a community of independent participants while globally minimizing costs (development and employment). Local efficiencies of point-to-point information design yield to the greater global efficiencies of the information environment in which a loose coupler operates. We investigate data sharing usage patterns and infospace environments within and across communities of interest and use these to identify characteristics of loose coupler design. We provide examples of loose couplers in both commercial and military environments. A loose coupler design methodology is discussed and we suggest opportunities for further research in this area.

Background

Net-Centric operations [1] are predicated upon loose coupling among participants. Loose coupling describes an architecture where integration interfaces are developed with minimal assumptions between two or more sending/receiving parties, thus reducing the risk that a change in one application/module will force a change in another application/module. [7] Each end of the exchange makes their information requirements known with minimal assumptions about other participants. From a Service Oriented Architecture (SOA) perspective, loosely coupled services can compose into more complex, composite capabilities. Such composites can be decomposed or reconfigured to derive additional functionality.

Interoperability requires that sending/receiving parties come to terms with their vocabularies so that both the

information producer and consumer have the same understanding of what the information means and how it is used. This means establishing and managing data definitions, schema and higher level knowledge captured in order to understand the context of the exchanged information. Business rules, how the information is used, may also need to be captured. Consequently, loosely coupled participants require a shared semantic framework to ensure messages retain a consistent meaning across the flow of information. Existing approaches to loose coupling have focused on the information and network interfaces, not on information design methodology.

Cursor on Target (CoT) is a strategy for enabling DoD systems to communicate much needed time sensitive position or "What, When, Where" (WWW) information. CoT leverages the ubiquitous XML technology and defines a common, terse yet extensible message format for communicating WWW information. CoT employs an extension data strategy similar to object oriented decomposition to manage and provide supplemental detail to the core WWW data. Using this approach CoT can be implemented in a variety of situations to easily and effectively represent Blue Force Tracking, Time Sensitive Targeting, mayday messages, Combat Search and Rescue reports, spot reports, ISR asset tasking, battlefield reservations, and many other tactical battlefield information exchange needs. [4]

Cursor on Target has been described as a *loose coupler* that provides a strategy to exchange a minimal amount of structured, high impact information in a variety of use cases among battlefield participants. In a general sense, loose couplers are information exchange structures that efficiently support the loose coupling approach. They typically comprise compact information objects that exhibit limited design richness, but with wide participant reach. [6] In this paper we provide an in-depth analysis of the loose coupler concept.

Information Sharing Usage Patterns

Many of today's information sharing environments typically comprise a heterogeneous network of participating systems, platforms, organizations, and users. They tend to act as more as "ecosystems" or

“communities” then an architected collection of components. These information environments continually change and evolve, participants come and go, functionality shifts with time, and small changes in one area of the “ecology” can create unexpected change in other areas. The behavior of the environment emerges through the types of participants, their interaction, and their relationships to one another.

Typical information sharing environments include:

Portals: Portals provide organized access to content and services on behalf of their users. Portals derive content and services from distributed sources, supply content to multiple platforms, and often function as collaborative workspaces.

Hub and Spoke Distribution: Hub and Spoke Distribution provides a central resource or location through which producers and consumers exchange information and services. The “hub” often provides data repository capabilities such as a central data base. The Joint Intelligence Operations Center (JIOC) and Combined Air Operations Center (CAOC) provide examples.

Value Chains: Value chains coordinate a network of multiple resources or participants in a (usually predefined) sequence or chain, the combination of the participants producing a capability through service orchestration. Each participant in the sequence adds “value” in achieving the capability. Typically, “raw” data is transformed into actionable knowledge through the participants in the process. Military mission threads such as close air support or targeting provide examples.

Peer-to-Peer: Peer to peer information sharing typically occurs over networks whose participants have minimal commonality in purpose and resources. Email and file sharing provide examples. These are often achieved despite significant interoperability obstacles in that all participants (peers) are equal from the network perspective.

Data “Usage Patterns” develop within information sharing environments. For example, data “routing” is a usage pattern typically found in Values Chains. Data “aggregation” and “querying” are usage patterns found in Portals. Hub and Spoke environments often entail request/response and scoring/ranking patterns. Typical usage patterns found in various information sharing environments include: Routing (nodal dissemination logic); Filtering; Querying; Aggregating;

Request/response; Sorting; Scheduling; Navigation; Synopsis/summarization; and Scoring/ranking.

Within a given information sharing environment, the information design methodology affects how successfully various usage patterns are supported. Information design methodologies should be evaluated as to how well they support achieving the target usage patterns. That is, given a usage pattern, the choice of one information design methodology over another may have broader applicability and lower implementation costs to environment’s participants.

Loose Couplers

Definition: A loose coupler is an information object design that optimizes data utilization (production and consumption) among a community of (semantically) independent participants to achieve global cost savings.

Loose couplers provide valuable knowledge benefits by enabling wide dissemination and usage. The local information design optimization of pair wise (highly integrated) data exchanges yield to the greater global efficiencies of the information ecosystem in which a loose coupler operates. In other words the utility/cost equation is globally maximized for all participants.

A particular value of loose couplers lies in their ability to work across heterogeneous participant information models. Loose Couplers are built upon common information that applies to a variety of circumstances. Thus loose couplers can be routed from node to node, service to service, across a distributed network or value chain, with different nodes performing different applications, and still deliver value-added content. Except for the core data, as loose couplers are routed through the network, the information contained within them can change based on the processing by intermediate or participating nodes.

An ideal environment for using a loose coupler is a distributed network in which there are a variety of heterogeneous participants, each of which can benefit in some way from the core data contained within. Loose couplers can be used in pair-wise applications as well, but with much reduced return on the investment.

Utility versus Cost of Information Exchange Adoption/Design

By definition, the value proposition of loose couplers is directly related to the breath of penetration achieved. The wider the adoption, the greater diversity that may be expected by participants in system heterogeneity, internal data models, architectures, interfaces, and programmatic (ownership, funding lines, configuration schedules).

Pragmatically, each participant must measure the utility (value) against the cost of adoption both in the short term as well as the long term. Seldom does any information exchange design get adopted by fiat. It is worth examining the factors that contribute to this analysis in general and the lessons that may impact loose coupler design.

Per economics, utility is a measure of the relative benefit gained from the consumption of a good or service. While cost is a measure of the value of the time, money, or labor expended to achieve a goal. The utility/cost ratio is a measure of the relative benefit gained for the given cost.

Participants, in considering the adoption of any information exchange standard, are faced with this benefit/cost analysis. On the utility side of the equation the adopter must consider factors such as:

- the expected monetary gain or savings from adoption,
- reusability beyond the specific business scenario
- increased transaction/processing efficiencies (e.g. transaction speed and volume),
- architectural fit,
- leverage of existing skills and assets,
- reduction of project risk (development, deployment, and maintenance),
- improved reliability, and
- adaptation to a variety of data usage patterns.

A variety of factors come into play when calculating adoption costs:

- Degrees of Freedom allowed in the information exchange design. The more degrees of freedom (DOF) or independent choices for variation, the more complicated the implementation. As DOF increase so do costs. DOF drive costs such as increased development time, potential for errors and additional testing, and performance penalties.

- Semantic and Syntactic correspondence. The further the information design and associated concepts are from the producer or consumer's internal business and data model the greater the cost involved in brokering the divide. Adapters and translation middleware increase costs.
- The knowledge required to make effective use of the information represented by the design. Additional education may be required for the developers, operational staff, or even the organization as a whole.
- Incremental costs to add new partners.
- Cost to track updates of the design over time.

The utility/cost analysis typically occurs independently for each participant. Consequently each participant strives towards a local optimum. Often the result may be adoption but partial implementation which decreases overall interoperability. The tendency to focus on local optimums also impacts information design strategy pursued through consensus-based standards development. Too often the result is a design littered with compromises (greater DOF) necessary to achieve consensus. Greater success is observed in more limited (bilateral) design activities.

In approaching the benefit cost analysis for a loose coupler, designers must balance the tendency for local optimization against large scale benefits. Aggregate utility/cost analysis must inform the design trade-offs in order to attain a more global optimum. Loose coupler design principals and techniques should be supportive of this type of analysis.

Another consideration is the practicality of how the global efficiencies are weighted in relation to various information sharing environments and the degree of control exerted. In value chains, control is more equitably shared versus that of a portal where the balance is weighted towards efficiencies of the aggregator/producer.

Loose Coupler Characteristics

Loose couplers provide value by incorporating or exhibiting characteristics that work to increase the utility/cost quotient for large numbers or users. We've identified the following characteristics that contribute to loose coupler construction or design. Some of these characteristics are related and thus overlap.

Minimalism: The data structure exhibits limited data richness that encompasses the common overlapping requirements of many participants.

Interrogation: A restricted format is used to support quick value assessment. Example formats include numeric and enumerated content. Tests for values within a range and for restricted lists are easily performed. Variable and deeply nested structures are avoided.

Categorization (taxonomy): A means to categorize the “what” of the data: its purpose, content, or identification. Taxonomies allow users to assess whether or not the information is of interest or to decide what actions need to be taken. The actual taxonomy (classification) may be described elsewhere.

Content/envelop duality: The core data in a loose coupler serves a dual purpose. First, it supplies key content that is passed from producer to consumer, or user to user. Second and somewhat more interesting is the core structure serves as a metadata envelop for the object. From this metadata, the core object can be used for routing, filtering, discovery, cataloging, and many other usage patterns.

Accepted components: The design employs existing widely adopted standards, such as ISO TIME or geolocation, and community practices. Using a widely adopted standard may decrease process processing costs.

Genericity: Techniques that can be used to address simultaneously multiple varieties of exchanges in many contexts. These adaptable forms may include support for polymorphic structures.

Extensibility: Mechanisms are provided to access additional data for increased value. Additional data may be provided by extension, reference, or service.

Immutability: The data structure integrity is consistent over time and any changes are fully backward compatible. Thus processing decisions do not have to be re-implemented.

Examples

We look at some DoD and commercial examples of information structures that exhibit characteristics of loose couplers.

Cursor on Target: CoT was discussed in the Background section of this paper. CoT works extremely well in value chain usage patterns such as situation awareness and time sensitive targeting. CoT operates in an ecosystem consisting of trusted peers with shared goals. Such structures may not work as well in commercial value chains where competitive pressures tend toward data protection and ownership.

OASIS Common Alerting Protocol/EDXL-DE: The Common Alerting Protocol (CAP) is a simple but general format for exchanging all-hazard emergency alerts and public warnings over all kinds of networks. CAP structures are designed to work across different warning systems, thus increasing warning effectiveness while simplifying the warning task. CAP messages are structured into segments: an <alert> segment, which may contain one or more <info> segments, each of which may include one or more <area> segments to describe the related geographical area. An optional <resource> segment provides a reference to additional information related to the <info> segment within which it appears. [9]

Dissemination metadata for CAP messages is provided by the Distribution Element specification. The Distribution Element “wrapper” is used to route CAP messages by including distribution type, geography, incident, and sender/recipient ID data. This XML structure employs the XML-based emergency Data Exchange Language (EDXL). The CAPS message(s) are included via the <contentObject> container element. [10]

Microformats: Microformats (www.microformats.org) are simple open data formats built upon existing and widely adopted standards. [2] Their intended purpose is to enrich the semantics of existing HTML structures without minimal impact – no change to existing tags – exploiting the tag’s optional class attribute. Microformats are modular structures that represent common knowledge and thus align to current human behaviors and usage patterns. As an example, the microformat hCalendar organizes and provides data on events. It can be embedded in current HTML documents that contain event data (subject, time, location) with minimal effort and impact. Microformat-aware applications, such as web browsers and services, can recognize and repurpose the microformat’s data via mashups.

Aggregate Value Raters (AVR): AVR’s are data structures that provide a quick but highly relevant assessment of value. AVR’s provide a single expression of a (often complex) computation of usefulness.

Examples of AVR's are eBay's feedback score, Google's PageRank, Morningstar's Star Rating of mutual funds, and Fair Isaac Corporation's FICO scoring model.

As an example, consider the eBay feedback score: eBay users like to deal with buyers and sellers that have feedback scores that inspire confidence. Feedback score represents the number of eBay members that are satisfied doing business with a particular member. Higher feedback scores instill confidence that a transaction will be completed successfully to mutual satisfaction. Buyers generally prefer to purchase from sellers that have attained some acceptable (to them) level of positive feedback through successful accounting and delivery of the item. Sellers prefer to deal with buyers who have shown their ability to pay quickly and with the minimum of hassle. These are reflected in the participants feedback score. However, this score is a computed representation of the aggregated success of many transactions with many buyers and sellers. The feedback score value is derived from its use by the millions of eBay account holders (high utility) and its simplicity in supporting transactional decisions (low processing costs).

Bartering (or Financial) Transactions: Bartering transactions routinely occur in many information sharing environments. Typical data formats include invoices, purchase orders, and financial statements. Similar data structures are found in retail portals (Amazon.com), producer/consumer hub&spoke (eBay), and value chain (RosettaNet). However, a bartering loose coupler seems not to have yet emerged, although such a loose coupler intuitively would have value across a number of usage patterns and a wide information ecosystem.

The following figure 1 shows how the above examples map to loose coupler characteristics. In some cases we were uncertain how well the structure complied with the characteristic definition. These are indicated by question marks.

	Minimalism	Interrogation	Categorization	Duality	Accepted Component	Genericity	Extensibility	Immutability
COT	X	X	X	X	X	X	X	X
CAP		X			X		X	?
EDXL		X	X					?
Microformats	X		?		X		X	X
AVR	X	X					X	?

Figure 1. Sample loose coupler characteristics matrix

Design Methodology

Figure 2 summarizes the considerations for loose coupler design across information sharing environments, usage patterns, and characteristics.

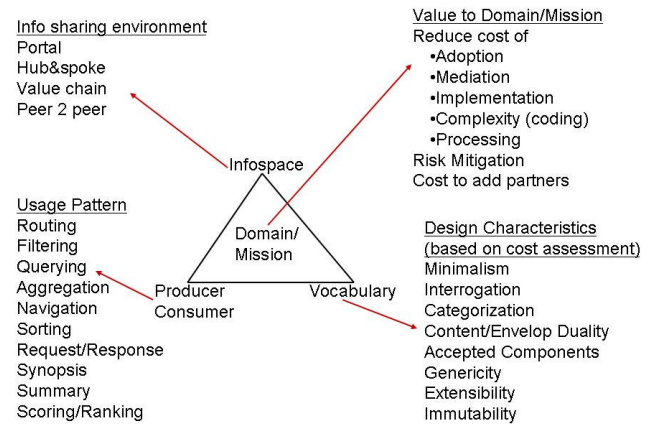


Figure 2. Loose Coupler Relationships

Loose couplers increase value globally through design principles that account for uncertainty and expansion to include new users and uses. Loose couplers help increase penetration by enlarging the information sharing environment in which they operate.

We have identified the following design principles for constructing loose couplers:

- Simplicity
- Optimize the Utility/Cost ratio community-wide
- Build on existing data standards in wide use
- Easy to interface to existing applications (easy adoption)
- Support multiple information environments and usage patterns
- Support categorization
- Extensibility (reference supplemental details)
- Promotion as a community standard
- Support web service implementations
- Componentization
- Backward compatible evolution
- Structural Integrity (extensions occur without structural change)

Extensibility in loose couplers is not generally achieved through changes to their structure. Extension mechanisms tend to be independent of the data structure itself. COT uses a core object design with the ability to include additional detail (extend the object) by including sub-

schemas. CAP structures are extended through expansion of the code space. That is, additional values are provided in the lists of codes to extend the use of the structure to provide additional data or functionality. Microformat structures are built from standardized components and are thus extended by inserting additional microformat components.

Conclusion/Areas for Further Study

Loose couplers show promise for information sharing in complex systems. These environments are characterized as having unpredictable behavior, fluid requirements, multiple competing stakeholders, and are susceptible to external pressures that can cause change across the entire system. [3] The US Department of Defense (DoD) net centric environment [5] is a good example of such a system, with many unpredictable external factors that often demand rapid response and flexibility to change [Albert]. Net centric operations for the DoD represents a shift from traditional system-based interactions toward information-based web transactions, adding the requirement for highly secure, reliable, and dynamic "on-demand" capabilities. [8]

Currently there seems to be a limited number of loose coupler examples. Current information exchanges are often optimized from a point to point perspective or at least within the confines of a specific use; e.g., interacting with specific web sites and web services. A major value of loose couplers lies in their ability to work cross domain and cross function or capability of supporting a variety of usage patterns. This organizing principle has not been fully explored or implemented.

This paper has introduced loose couplers as a design methodology that bears consideration as we move to large scale information sharing that requires support for critical interoperability across heterogeneous participants. We suggest that a strategy for building and using loose couplers needs to be fully developed.

Future study on loose couplers should consider:

- Effective implementation of loose couplers in various formats
- Refinement of the loose coupler characteristics presented in this paper
- Metrics for evaluating techniques that influence utility and cost
- Adoption strategies that recognize each participant utility/cost calculus. For example,

compensation for participants that incur higher costs in implementations or development of software developer kits to promote adoption

- Lessons from economic theory as they apply to loose coupler employment
- Information sharing environment and data usage pattern's affect on design strategy
- The effect of the degree of control exhibited in the information sharing environment on design.

References

- [1] D. Albert, J. Garstka and F. Stein, Network Centric Warfare, CCRP, 1999.
- [2] J. Allsopp, Microformats: Empowering Your Markup for Web 2.0, friends of ED, March 2007
- [3] Y. Bar-Yam, When Systems Engineering Fails--- Toward Complex Systems Engineering, 2003 IEEE International Conference on Systems, Man & Cybernetics, October 5–8 Washington, D.C., USA, 2003.
- [4] Cursor on Target, Military Information Technology Online Edition, vol. 8, issue 7, September 2004.
- [5] Department of Defense Net-Centric Data Strategy, ASD/NII, 9 May 2003
- [6] P. Evans, T.S. Wurster, Blown To Bits: How the New Economics of Information Transforms Strategy, Havard Business School Press, 1999.
- [7] D. Kaye, Loosely Coupled, The Missing Pieces of Web Services, RDS Press, 2003.
- [8] R. Miller and R. Cherinka, Engineering a Complex Information Enterprise: A Case Study Architecting the Department of Defense Hourglass, International Conference on Enterprise Information Systems and Web Technologies, July 2007
- [9] OASIS Common Alerting Protocol, v.1.1, October 2005
- [10] OASIS Emergency Exchange Language (EDXL) Distribution Element, v. 1.0, May 2006