

Executive Overview



**business
transaction
information
management**

Why do Businesses Need CAM?

Today's business interactions are becoming increasingly sophisticated, and the legal and social impacts more pronounced. It is no longer enough to simply exchange basic information with few or no edits and checks involved. Further more to gain maximum business advantage requires flexible information flows that can react rapidly and effectively to market changes and opportunities, not to mention formal service requirements between partners. Some examples include: being able to adapt to cross-border shipment regulations to ensure timely deliveries, tracking of goods and services, ensuring appropriate delivery methods (refrigeration, fragile goods, perishable goods, security of goods), through to parts and supplies meeting exacting engineering and technical requirements to ensure the correct part or service is provided. We need only to mention the recent Mad Cow disease event to realize how important all this may suddenly become, and then think about potential threats to supply chain providers in today's volatile world to realize the degree of importance accurate information now has.

To address these needs information exchanges are moving to technical formats using XML technology worldwide. However XML by itself is only a markup language, it was never intended to support exacting business interchange definitions, rules and industry vocabularies. To provide that extra level of robust information definition and exact control the Content Assembly Mechanism (CAM) has been developed as an OASIS open specification and toolset. Business users can quickly and easily use CAM templates to declaratively assert these missing business rules and structural information requirements.

The Issue of Context in Business Interchanges

Particularly important is the ability of CAM templates to enable the use of *business context*. Knowing the context of any information exchange is critical. For example a seller may need to know such things as: is this shipment urgent, is this a preferred customer, do they need English or Spanish, what model is the part for? Without context mistakes and run-on costs rapidly ensue.

In automating information integration, knowing and defining context of use is the single most pervasive and important factor. The older EDI systems however have no way to support dynamic context driven content assembly whereas CAM provides this better than any other XML syntax system today. In fact CAM templates work for both old EDI payloads and new XML based transaction structures.

CAM provides a rich yet simple context mechanism defined in XML syntax that enables interactions to be tailored by specifying the context parameters (as in the seller example above) and the appropriate agreed control values they will contain. There are also guidelines toward determining roles and methods to deduce what these appropriate significant parameters should be. The concept here is "*no surprises*" which means that users of CAM templates know exactly what to expect, and why, from their business partners because they can inspect and agree to the context parameters before hand.

This context mechanism is at the heart of why CAM is so effective. By allowing users to quantify what their context factors are precisely, this removes the guess work from business transaction exchanges between

partners and allows them to formulize their collaboration agreements exactly. It also makes re-use and identifying potential candidate CAM template models much easier. Simply knowing what the external parameters are can instantly qualify if a template makes sense for a user and allow them to therefore select it.

Relating to this is the further need in today's interconnected world to be able to leverage industry dictionaries of standard pre-defined information elements. CAM directly supports the use of registries of industry vocabulary definitions for content references within business transactions. Therefore information exchanges can be assembled from pre-built proven industry components and structure definitions.

By using CAM technology businesses can secure and dramatically improve the quality and accuracy of their information exchanges. The business benefits then accrue through the ability to handle more business partners needs, more types of products, more sophisticated products, while shortening the time to deploy by eliminating production errors and making pre-production testing simple and complete.

Having established the business needs, the remaining questions concern what the required software components to acquire CAM technology are, how do these work, how does that fit with your existing infrastructure, and what is involved in adoption and implementation?

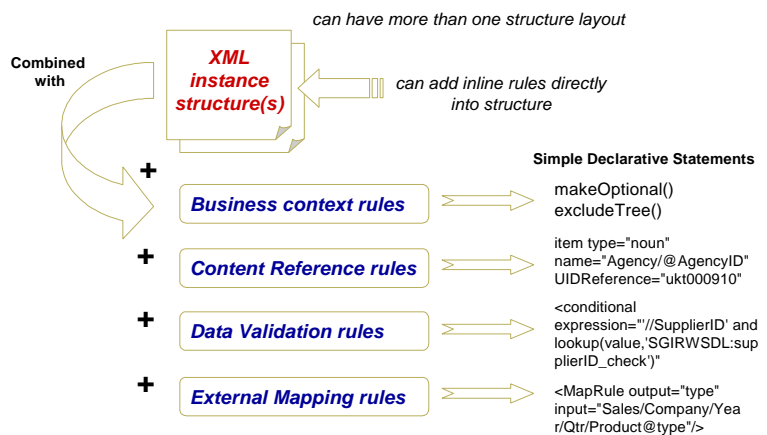
CAM Technology Overview

The CAM specification provides an open XML based system for using business rules to define, validate and compose specific business documents from generalized schema elements, attributes and structures.

A CAM rule set and document assembly template defines the specific business context, content requirement, and transactional function of a document. A CAM template must be capable of consistently reproducing documents

that can successfully carry out the specific transactional function that they were designed for. CAM also provides the foundation for creating industry libraries and dictionaries of schema elements and business document structures to support business process needs.

The figure here shows a summary of the technical components that a CAM template can contain. They can be used altogether or in combination. The structure instance section is required; the remaining four are optional, so a CAM template can be as simple or sophisticated as the business needs dictate.



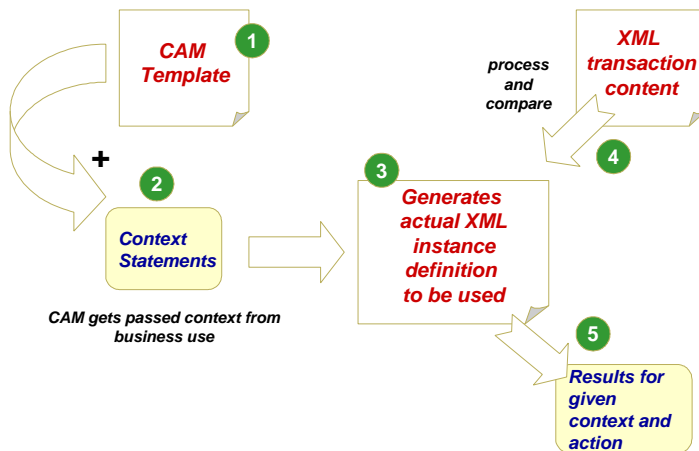
Next we describe how to use CAM with typical production environments today such as Microsoft's BizTalk™ Server and .NET environments, or other transaction processing systems just as the Sun J2EE based environments.

Using CAM template processing

Integrating CAM processing into an existing environment is simple and straightforward. First you need to determine what role or roles you wish CAM to perform. A typical role for an initial adoption is using the CAM processor to validate inbound XML transactions in a processing queue based on who the originating sender is and the context of that transaction.

An incoming transaction is selected, its profile referenced to find the appropriate CAM template type for that transaction, and then a context parameter set file may also be optionally selected from that senders profile.

The CAM processor then reads in the XML transaction, loads the CAM template and the context value file (if provided) and then validates the information and structure in the transaction by comparing the rules in the template with the information in the actual XML transaction record(s). The figure here shows the steps involved in CAM processing.



Importantly CAM is able to handle very specific structure variances and rules, such as knowing when a customer number or part type is needed or allowed depending on the type of item that is being ordered. All this is encoded using simple and clear XML declarations in the CAM template itself.

CAM can also do lookups of business information using the template lookup rule mechanism. This can check information against backend application database information such as part numbers, stock on hand and other volatile information that cannot be written as a static rule in the CAM template itself. If CAM detects a problem it can direct error responses accordingly, along with appropriate error messages that have been defined in the CAM template. This allows highly specific and accurate problem reporting, rather than vague and non-specific reactions to be returned back when a problem is encountered.

The CAM processor itself is a self-contained component that can either be called directly by the existing transaction processing software, or invoked as a web service. In

either case it is provided with parameters that point to the XML transaction to be processed and the CAM template and context details that apply.

Another excellent role for CAM is as a design time tool, allowing business partners to pre-validate sample business transactions before they use them in production. Therefore in this case you can provide CAM templates to your partners so they can pre-test their own transaction implementations. In this case they would run the CAM processor locally on their own systems, or remotely as a web service from your own network facilities.

The next role is in automated transaction processing systems such as a business process engine (BPEL or BPSS) and it may use a CAM processor to direct the creation or processing of business transactions for it. In this case a step in the business process will have a CAM template associated with it that will be invoked. The CAM processor can then produce the desired information processing for the business process engine.

Considering this business processing picture CAM processing can be used to assemble or re-format transactions either outbound or inbound. This capability is enabled by the merge and mapping features of the CAM template. Since the CAM template allows you to specify more than one transaction structure, you may decide to have one structure for the information coming from your application databases and then another structure for the resulting output layout of the transaction you require to send. In this case the CAM processor will determine the exact layout structure dynamically by apply its context assembly rules, including or excluding structure pieces accordingly. For example if an order contains perishable items, those items require handling and storage instructions to be added to the transaction structure itself.

Once the target structure is assembled, then the CAM processor will complete the transaction itself by merging and manipulating the information from the raw input data structure. Again these rules are expressed as simple declarations using XML syntax and

predicate commands within the CAM template itself.

Obtaining CAM processing

An open source CAM processor is available that supports validating transactions with a CAM template. The CAM templates themselves can be created using any simple XML text editing environment. The first thing you require when constructing a CAM template is an actual XML model of the information structure you wish to manipulate. You can either enter this directly from scratch into the CAM template, in XML, or you can cut and paste an example from some XML that you are already using¹. Many samples are available that serve as illustrations for various scenarios and industry transactions.

Since the CAM processor itself is designed to be highly interactive with feedback messaging, this makes it also a good learning tool for staff creating their own CAM templates. The CAM specifications themselves also contain helpful samples and use guides throughout. There is also a CAM template model available in the commercial VisualScript editing tool product that will automatically create a baseline CAM template. A 30 day free trial version is available for those wanting to try this out.

Unlike complex XML syntax approaches, CAM templates are designed to be open and easy to read and build by hand for human designers. The target transaction structures are specified directly in XML itself. This means no special knowledge is needed; designers can simply type into the CAM template the exact structure layout they wish to use. This is WYSIWYG for XML structures.

Designers can also include sub-structures command that insert from pre-built components too, allowing commonly used patterns to be made consistent across a range of content assembly structures.

¹ Note: a tutorial of the technical steps to create CAM templates is provided in the companion “*CAM Technology Overview*” brochure available from the OASIS CAM TC documents center.

Creating Industry Catalogues

Since CAM templates are just simple XML files themselves they can also be readily and easily shared within an industry group or a business enterprise. They can be stored into an XML Registry or traditional online cataloguing system and then searched for and located accordingly. Whole libraries of CAM templates can be built for popular and typical industry transaction formats, such as the OAGi BOD formats that are used extensively by ERP systems or financial CAM templates for common invoice and popular financial transaction structures. Users then simply download the appropriate CAM template and set their own CAM context variable values to customize it for their own use.

Industry Dictionaries and Vocabularies

Built into the CAM templates is the content referencing system. This allows developers to create libraries of pre-defined elements with characteristics, facets and business rules associated with them automatically. These definitions can either be specified directly in-line into the CAM template, or can reside in an industry registry of terms and structures.

Additionally by using the content referencing system the information crosswalk between different industry structures can be automatically deduced and documented.

The content referencing system is very flexible and supports a wide variety of registry interfaces directly. It also allows industry groups to use their existing dictionary elements and code systems directly, without requiring invention of new codes or details. Best of all, existing legacy transactions that are in use today can simply be upgraded and made CAM-aware without any changes to those wire transaction formats themselves. By creating a CAM template for any such transactions and then completing the content reference section this will automatically provide the crosswalk to the industry dictionary definitions.

When used in an environment where an industry dictionary is not readily available the content referencing system allows

developers to quickly construct that and classify those element details.

Local language decoupling

A further huge benefit of the content referencing system is locale and language usage decoupling. Often XML implementations become mired in arguments relating to the exact XML tag names to use for content, and also the information content itself requires localization into another language. This often involves conflicts around the use of elements or attributes or both within the XML structure since XML itself does not provide you with a single consistent way to do this (attributes of attributes are not permitted, and elements cannot be enumerated and tokenized).

The content reference system frees you from these constraints by allowing you to specify a reference code for an element or attribute directly. You can therefore apply the rules you where you need them. Therefore the CAM processor can consistently apply rules and manipulations regardless of the actual structure layout, element or attribute name used. From the CAM perspective the elements `<process_time/>`, `<Pdatetime/>`, and `<madewhen/>` are all functionally identical when they all have the same content reference identifier associated with them.

By providing the locale language definitions or these items as well, then you have the ability to read in codes in one language and then output them directly to a different one.

Who uses CAM templates?

In this segment we provide examples of a selection of our current audience set and people who are working with and uses of CAM templates today:

- Developers who are working with handling XML transaction content
- Business analysts creating industry domain transaction standards
- Engineers working on smart applications and agent applications with XML
- Business process engineers defining workflow interactions

- Information modelling analysts creating dictionaries and vocabularies
- Production managers running high volume eBusiness hubs
- Mobile device engineers working with rendering content to small footprint displays
- RFID engineers needing to link XML to physical characteristics of items and build catalogues that can be automatically referenced
- Consultants advising customers on optimal ways to deploy XML technologies.

The following items next provide samples of possible use case for CAM processing.

A CAM processor can be deployed as a web service to allow business partners to pre-validate XML instances before using them in message exchanges. By presenting an XML instance to the web service and selecting a CAM template the partner can have returned a detailed report about the results of the tests and their outcomes.

Within industry standardization efforts CAM templates can be used to capture all the design rules that apply to an industries use of XML structures. The CAM processor can then provide a validation and conformance service. Industry groups can also create CAM templates to document older legacy transaction formats and provide XML based semantics and usage rules for those.

When integrating with E-Forms systems the CAM processor can be used in a variety of modes. It could create forms given some input business transaction, or it could take the output from a form entry process and then validate it and re-structure the content and deliver it as a business transaction.

In short the CAM processor can be used in a variety of roles wherever manipulation or validation of information content structures is required.

A Real World CAM Scenario

The original implementation of CAM by British Telecomm (BT) in the UK is providing transaction validation against hundreds of trouble ticket message formats that BT receives daily. These trouble ticket formats

used in the telecommunications industry are XML messages using the xCBL syntax approach and are highly complex XML structures. The information carried in them is highly variable and differs according to the originators own telecommunications switches and configuration details. The message details change frequently every three to four months as new services and products are introduced and provisioned in response to demands from highly dynamic and technology driven markets within the industry.

Managing these structures and formats takes significant manpower and especially to detect when solution partners have made changes to their transactions and details without first notifying the systems connected to them. This may also happen inadvertently when new devices are installed in the field and they start producing new trouble ticket reports automatically.

Being able to direct and manage this implementation space using the XML scripts in the CAM templates significantly reduces this maintenance burden and also allows changes to be implemented rapidly and easily.

Partners can also begin to become involved in the process themselves by managing and maintaining the CAM template definitions that relate to their own production systems and configurations. They can then share those templates with their partners when changes are about to be placed in production.

How Does CAM Work With or Compliment Other Semantic Definition Initiatives (UDEF, UBL, OAG, RosettaNet, HL7, STEP, eprXML)

The CAM technology has been designed fundamentally to leverage dictionaries of noun definitions and libraries of transaction formats built from industry initiatives. By providing the content reference section in the CAM template the processor is able to automatically retrieve semantics from industry registries about individual elements in a XML transaction. This allows consistent definitions to be deployed across a set of industry business transactions. It also includes the ability to version and sub-version definitions. This is especially important for the processing of code lists. Code lists

provide the basis for up to 50% of information flowing in legacy EDI transactions. Industry groups define extensive code lists for the products and services used by their members. Yet the W3C XML schema system does not support code list definitions directly. Therefore CAM provides a key capability with its built-in *lookup ()* functionality.

Industry initiatives today have created dictionaries of their nouns (elements) and verbs and labelled them with reference codes. They have also provided classifications and taxonomies for these as in the case of UDEF particularly. By allowing these industry groups to load their dictionaries into XML instances in a registry the CAM approach unlocks the potential of these dictionaries to effect significant improvements in interoperability of information in those industries.

Providing the means to assemble consistent transactions from pre-defined libraries of structure components and also to create libraries of business process definitions that can be context driven are key functionality that CAM delivers.

By linking a CAM processor service to the registry these industry groups can then allow members to validate XML transactions against the standard definitions in their registries or download CAM templates and associated business process definitions that they can use directly in their own systems.

By combining CAM technology with a registry in this way provides the foundation for even more sophisticated semantics in the future. Taxonomies and ontologies are emerging as ways for machines to understand more about knowledge in a way that today only humans are able to. Registries can provide ontology and taxonomy driven search results. This allows agent software and human researchers to more effectively locate business transaction processes, such as CAM templates, and determine if they are suitable for the task they need.

Enhancing the work of industry groups is a significant benefit that CAM technology delivers today.

CAM can also be used in tandem with UML. UML does have a production rules system that can be used to output XML from UML diagrams and it is anticipated that this can allow UML models to output partially complete CAM templates that can then be hand-edited in XML to complete a full information model.

Perhaps one of the most exciting potential uses for CAM is with the emerging OASIS work on EPR (Electronic Process) specifications for service oriented applications². Since these are required to be fully model driven, having the CAM templates able to resolve dynamically through context parameters the information exchange formats allows designers of eprAPL (EPR Application) solutions a rapid and consistent way of implementing these.

Summary

Reviewing what CAM represents and the opportunities it opens up we find a range of use cases; from sophisticated discreet tasks as part of specific architectures such as ebXML business processes or Service Oriented Architectures environments, to simple local document verification as a standalone business tool.

The value of the CAM approach is in managing context and business rules directly and tying these to consistent information semantics. The lesson learned is that consistent information exchanges must include context management. The result from this will be to transform the ability of industry to deploy successful and interoperable eBusiness information exchanges simply and quickly.

Contacts and Additional Information

The CAM technology is a product of the OASIS Business Content Assembly Mechanism Technical Committee. More details can be found from the specifications and documents that the committee has produced. These include the latest CAM specification, presentations introducing CAM, and examples of CAM templates. These and other documents can be obtained through the OASIS website at: www.oasis-open.org.

For more information on how to participate in CAM activities, please see the membership links from the main OASIS web site or the CAM TC area directly.



Chair: David Webber – drwebber@acm.org



² For more information on the EPR work and the eprXML and eprAPL components – please see the OASIS BCM TC area at <http://oasis-open.org>