Use of XML in the Design and Specification of a new High Assurance Controlled Interface

Joe Loughry
Lockheed Martin IS&S

2004 April 06

Abstract

This paper is part of a series of reports intended to give an overview of experiences so far in the specification and construction of a new high assurance controlled interface in the EAL6 or 7 region. Significant milestones achieved to date include use of the Single XML Description to reconcile multiple Certification & Accreditation (C&A) standards, and demonstration of completely automated, hands-off generation of C&A documentation. The innovative use of a Single XML Description has proved to be sufficiently expressive even when required to handle highly abstract operations and Formal Methods notations using non-Latin alphabets. The ease of parsing XML has facilitated the use and reuse of existing text processing tools in ways not originally envisioned.

1 Introduction

This paper is part of a series of reports intended to give an overview of experiences so far in the specification and construction of a new high assurance controlled interface in the EAL6 or 7 region.

No currently available Cross Domain Solution exists that is capable of meeting DCID 6/3 Protection Level 5 (PL-5) requirements or NIAP assurance levels higher than EAL4. The need for a high assurance controlled interface suitable for use in tactical, deployable, and multinational coalition environments has long been recognized—alongside the glaringly obvious lack of actual, existing products capable of meeting NIAP assurance levels higher than EAL4.

What is described in this and the following series of papers and progress reports is the design and construction of an appropriate solution to the preceding problem. The proposed solution must fit the needs of the present global environment. It must provide the greatest achievable level of protection for classified information; it should come with certification by trusted authorities that it is suitable for protecting multiple enclaves, all the way from unclassified/foreign releasable through sensitive compartmented information (SCI). It must be flexible, extensible, and applicable to the widest possible range of

throughput, data formats, diverse networking protocols, and environmental conditions. It must be reliable in the presence of hardware and software failures. It must be content-agile, deployable on short notice, able to handle complex and shifting multinational coalitions. It must be affordable, and scalable, and play well with others. It should leverage the effort and knowledge already expended on previous development.

2 High Assurance Controlled Interface (HACI)

For tactical flexibility, the Lockheed Martin High Assurance Controlled Interface (LM-HACI) is designed with remote maintenance, small form factor, rough environmental conditions, and remote rule set update capability in mind. It is designed to leverage previous investment in the large existing library of prewritten MAG format specifications already in existence. To maintain high assurance without placing unnecessary demands on the rest of the infrastructure, pluggable Type 1 encryption will be used. This effectively gives us digital signature capability for free, in addition to facilitating the delivery of rule set updates over untrusted communication channels.

Software updates or rule set changes will be digitally signed by NSA after the developer has validated and delivered the material to NSA. In the absence of a valid digital signature, LM-HACI will not accept an update.[1] Rule set updates are delivered as an opaque package. Without the right crypto keys, to the person who has managed to intercept an update in transit, the package is nothing more than random-appearing bits. Even to another LM-HACI, if it is not specifically addressed to that box, the remote update is a completely opaque package. In this way we hope to implement a practical remote maintenance capability for fielded systems that is acceptable under DCID 6/3.

The LM-HACI is being designed to run on a range of COTS hardware. Board support packages and high-assurance process management interprocess communication functionality are provided by use of the INTEGRITY-178B RTOS [7] from Green Hills, Inc. Running under the control of the safety-critical INTEGRITY-178B executive will be the MILS Architecture [5] controlled interface functionality that is the subject of this series of reports.

2.1 Formal Methods

Software intended to provide the necessary level of information assurance (IA) must be developed using Formal Methods [3, 6, 10]. Derived from the highly successful MAG parser/formatter used in Radiant Mercury, the Controlled Interface software component of LM-HACI is developed with the rigor of a mathematical derivation. It is equally subject to review, refutation, and logical proof. Each step in the transformation from requirements through formal specification

[1] PKI issues in the field, especially the certificate revocation problem, will have to be resolved before this can become a reality.

to executable code is defensible, defined, and limited. The following sections describe progress to date in the development of software, with particular attention to the use of XML and Formal Methods.

3 Milestone I

The first task was to produce a Common Criteria (CC) Protection Profile defining the broad outlines of requirements for security functionality and assurance levels for LM-HACI. An algorithmic approach was followed, first by mechanically extracting requirements from the CC (version 2.1) documentation; next performing a straightforward transformation into XML; thereafter developing a capability to filter any desired Protection Profile mechanically out of the baseline XML data store. Milestone I was intended as both a demonstration and a prototype of the methodology to be used later in the process of constructing LM-HACI. We shall see later some implications of design decisions made at this early point in the development process.

An *ad hoc* XML schema evolved throughout the process of encoding Parts 2 and 3 of the CC into XML. In the absence of an existing XML schema or DTD, at the urging of some experience users of XML, it was decided to expend the additional effort to produce an XML schema for the project. That effort is currently ongoing.

3.1 Common Criteria into XML

The CC consists of a mixture of at least two distinct types of information, for the most part easily distinguished. The Introduction, most of the explanatory material, and parts of the appendices are written in a relatively informal style, which we call "free text" for the purposes of this paper. *Security Functional Requirements* and *Security Assurance Requirements* are written in a formal style that we call "formatted text." Other portions of the CC such as Application Notes, some Tables, and parts of the appendices are written in a mixture of formal and informal style. These had to be dealt with on an individual basis.

Free text portions of the CC were not explicitly converted into XML except for bracketing these passages between `<freetext>` and `</freetext>` tags for ease in processing.[2] Formatted text portions of the CC were manually encoded into XML with a granularity sufficient to exactly reproduce the original typographical formatting of the CC, with the exception of page numbering and some non-semantic line breaks. Typographical cues were converted to semantic encoding in XML according to the *ad hoc* XML schema. In the course of manually encoding all of the formatted text, the XML schema evolved significantly to fit, with changes back-propagated as necessary to maintain consistency.

[2]Future work may include a more fine-grained parse of portions of the CC contained in `<freetext>` blocks, with the intent of being able to completely regenerate the original CC documents in their entirety from the Single XML Description. This is expected to greatly facilitate maintenance of the Single XML Description as updates and changes to the CC occur.

Some of the tables in the CC contained critical information that does not appear elsewhere in the text and consequently they were converted into XML as well. The resulting XML files were processed through an XML validation utility [2] repeatedly until no more errors were found.[3]

3.2 DCID 6/3 into XML

Director, Central Intelligence Directive 6/3 [4] was similarly encoded into the Single XML Description. The CC is unclassified; however, DCID 6/3 is marked FOUO. Special semantic markup was therefore indicated to express classification information directly in the Single XML Description. This turned out to be useful later on, as it facilitated the automatic production of unclassified documentation when needed.

A common format for requirements was beginning to emerge (see Figure 1).


```
<requirement tag="Audit9">
  <node tag="e90b29a5bb26606a0a3cc2e0fe04e5f6"
        description="Audit9">
    <requirement_text>
      Auditing procedures, including:<itemize>
        <item>The capability of the system to
          monitor, in real-time, occurrences of,
          or accumulation of, auditable events
          that may indicate an imminent
          violation of security policies.</item>
        <item>The capability of the system to
          notify the ISSO of suspicious events
          and taking the least-disruptive action
          to terminate the suspicious events.
        </item></itemize></requirement_text>
  </node>
  <arc begin="e90b29a5bb26606a0a3cc2e0fe04e5f6"
       end="e97c4c9bb65a49a7f567ed3f6b1818bd"/>
</requirement>
```

Figure 1: Example DCID 6/3–CC mapping showing, to the left, one possible graphical notation and to the right a portion of the Single XML Description. The 32-digit hexadecimal values are random identifiers selected mechanically during the conversion to XML and do not encode any specific information.

4 The Single XML Description

The Single XML Description was intended to carry the entire project from requirements all the way through deployment of LM-HACI to field users. In the course of developing the Single XML Description schema, a DCID 6/3–CC mapping was naturally produced.

[3]Several minor errors in the CC were found and corrected during this process. Locations of corrected text in the CC portions of the Single XML Description are indicated by special semantic markup.

Requirements traceability in the Single XML Description is indicated by means of arcs and nodes, facilitating use of familiar theorems and useful transformations from graph theory [1]. Initially, use of the XLink [11] recommended standard for XML linking was considered, but it was rejected as being unnecessarily complicated.[4] This design decision may have to be reconsidered later.

Instead, the Single XML Description schema was extended to define syntax and semantics for a generalized linking scheme that could support location or region nodes and directed arcs.

4.1 Link Syntax

"Location" nodes are indicated in the XML data stream by `<node/>` tags as in `<node tag="..." description="..."/>` and define a particular byte offset within a sequential file. "Region" nodes bracket a contiguous stream of text between `<node tag="..." description="...">` and `</node>` tags and may be arbitrarily nested (see Figure 2).

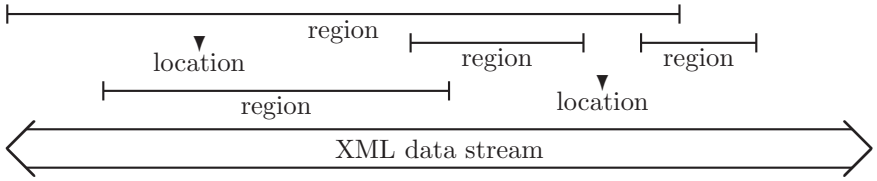

Figure 2: Location and region nodes may be arbitrarily nested.

For simplicity, only 1:1 directed arcs are defined. While a bijective CC–DCID 6/3 mapping is desirable, at this time it does not appear to be practicable. Instead, surjective and multivalued relationships are expressed using multiple arcs, analogous to additional rows in a relational database. It is anticipated that this simplification may require revisiting in the future. For now it has proved adequate to the need.

5 Milestone II

The second project milestone was to show that required documentation could be generated mechanically when needed from the Single XML Description. For the purpose of this milestone, the draft *Protection Profile for High Assurance Controlled Interfaces* [9] was chosen. While not all of the functional requirements of the envisioned LM-HACI actually existed in the Single XML Description yet, it should still be possible to generate the entire PP, in draft form, completely hands-off in its final presentation form. The process is somewhere between the operation of a compiler and the execution of a complex query language against

[4] The principal reason why XLink was felt to be unsuitable is that the current XLink standard requires outside-accessible URI links, which are not available in the isolated development environment required for LM-HACI work.

an existing database. As new content is added to the Single XML Description, an up-to-date PP can be regenerated at any time.

5.1 Draft Protection Profile

A Protection Profile describes a set of end-user or purchaser requirements for an information system [3]. Existence of a PP is not required for certification of products under CC or NIAP, however; the Security Target of a product in evaluation may optionally claim compliance with one or more Protection Profiles, or none, if no applicable PP already exists. A well-defined PP serves as a useful benchmark for customers searching for an IT product that meets their needs. At the time of this writing, there is no existing PP for DCID 6/3 Controlled Interfaces at Protection 5 and Evaluation Assurance Levels above EAL4+Flaw Remediation. Consequently, it was necessary to write one.

The structure of a PP is similar to that of a Security Target and is defined in Part I of the CC [3]. Much like the CC and DCID 6/3, the PP contains stretches of both informal and formal text; for this reason all of the content of a PP should fit easily within the XML schema developed for the Single XML Description. In the interest of expediency, informal sections of the PP (front matter, Introduction, Target of Evaluation (TOE) Description, descriptive text, and Rationale) were encoded as `<freetext>` regions, with occasional special semantic markup introduced as necessary to provided needed cues to the formatting process. The largest two sections of the PP (*Security Functional Requirements* and *Security Assurance Requirements*, respectively) would be extracted from the Single XML Description during the "compilation" process (or complex database query) that comprises the bulk of the PP generation process. Tables and figures, including front matter and back matter components, would be automatically generated by the formatting system.[5] Appendices, as before, would be handled specially on an individual basis as necessary.

5.1.1 Formal Methods Notation

One of the appendices of the PP immediately showed the wisdom of choosing XML early on as the *lingua franca* of the project. The LM-HACI is to be certified at EAL7, the highest possible level of assurance, and therefore has to be designed and constructed using Formal Methods. Many different Formal Methods notations exist; few limit themselves to the familiar ISO Latin-1 alphabet. The Formal Methods notation chosen for Milestone II was Z ("zed"). Specifications and proofs in Z require a number of special symbols and typographical formatting conventions in order to be properly expressed (see Figure 3).

Some new LaTeX macros and environments were created to properly format stretches of Z in the PP, after which it remained only to define suitable tags in the XML schema to be able to encode Z expressions, formal specifications, and—eventually—proof of correctness in the Single XML Description with sufficient flexibility and expressiveness to handle the language. As before, the XML

[5] LaTeX a document preparation system

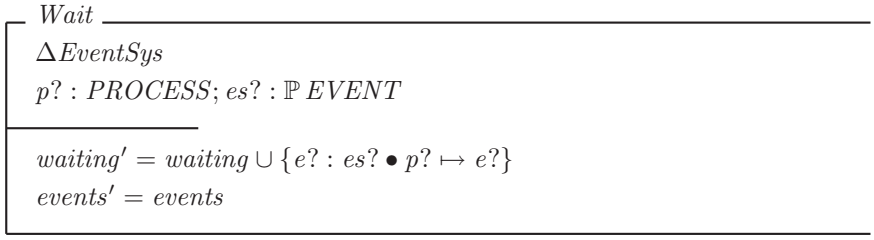

Figure 3: Example of a formal specification in the Z notation (after [8]).

schema evolved during the process to fit. A portion of the Single XML Description corresponding to the Z specification in Figure 3 is shown in Figure 4.

```
<spec type="formal" notation="Z" tag="62091a78da0d72c9690a3d5b9d97cf33">
  <schema name="Wait">
    <decl>
      <changeofstate name="EventSys"/>
      <input name="p" type="PROCESS"/>
      <input name="es"><type>
        <powerset><basictype name="EVENT"/></powerset></type></input>
    </decl>
    <pred>
      <eq><lhs><prime><id name="waiting"/></prime></lhs>
        <rhs><union><lhs><id name="waiting"/></lhs>
            <rhs><set><compose>
                  <lhs>
                    <function><lhs><input name="e"/></lhs>
                    <rhs><input name="es"/></rhs></function></lhs>
                  <rhs>
                    <maplet><lhs><input name="p"/></lhs>
                      <rhs><input name="e"/></rhs></maplet></rhs>
            </compose></set></rhs>
        </union></rhs></eq>
      <eq><lhs><prime><id name="events"/></prime></lhs>
        <rhs><id name="events"/></rhs></eq>
    </pred>
  </schema>
</spec>
```

Figure 4: Corresponding representation in the Single XML Description of the Z schema in Figure 3. (The representation is conceptual.)

6 Summary and Conclusions

The Single XML Description has proved to be a fundamentally sound tool for use in developing LM-HACI. While we have hardly plumbed the depths of its usefulness yet, XML has shown itself to be sufficiently expressive for our needs, even when required to handle highly abstract operations and non-Latin alpha-

bets. The ease of parsing XML has facilitated the use and reuse of existing text processing tools in ways not originally envisioned. The MAG parser/formatter, for example, has proved extremely capable for slicing and dicing XML descriptions of its own behavior, essentially employing the tool itself in the fabrication of a new tool.

The Single XML Description will be employed throughout the system life cycle, and it is expected that it will be incorporated into the next generation of LM-HACI. What is certain at this time, however, is that the accumulated knowledge from LM-HACI development effort, because it is written in XML, will not suffer from the problem of "bit rot" that affects the historical experience value of too many projects.

References

[1] Bela Bollobas. *Modern Graph Theory.* Springer–Verlag, New York, 1998.

[2] Scholarly Technology Group Brown University. XML validation form. URL: `http://www.stg.brown.edu/service/xmlvalid/`.

[3] Common Criteria Project Sponsoring Organizations. *Common Criteria for Information Technology Security Evaluation*, August 1999. CCIMB-99-031, Version 2.1.

[4] Director, Central Intelligence. *Protecting Sensitive Compartmented Information Within Information Systems*, August 2000. DCID 6/3.

[5] Michael R. Dransfield, W. Mark Vanfleet, Steve Grimaldi, Lee MacLaren, Jahn A. Luke, and Ben A. Calloni. Deeply embedded high assurance (multiple independent levels of security/safety) MILS architecture, July 2003.

[6] David Gries. *The Science of Programming.* Springer–Verlag, New York, 1981.

[7] Green Hills Software Inc. URL: `http://www.ghs.com/products/safety_critical/integrity-do-178b.html`.

[8] Jonathan Jacky. *The Way of Z.* Cambridge University Press, Cambridge CB2 2RU, UK, 1997.

[9] TBD. *Draft Protection Profile for Automated Guard and Sanitizer in Environments Requiring High Robustness*, 2004. Draft version 0.298.

[10] Mark Vanfleet and Michael R. Dransfield. Design guidance for the MILS architecture to provide MLS damage limitation. Draft No. 1 (NSA Information Assurance Directorate, September 2003.

[11] World Wide Web Consortium. *XML Linking Language (XLink) Version 1.0*, 2001. W3C Recommendation 27 June 2001.