

CMIS: In Search of a Common Ground

David Choy

Chair, OASIS CMIS Technical Committee

Standardization is often viewed as a sign of maturity for a technology segment. There is some truth in this. Pioneer technologies are inherently proprietary. As demand grows, a multitude of competing technologies arise, giving legitimacy to a new industry. While success creates opportunity and excitement, the lack of interoperability between competing technologies eventually leads to separate pockets of technology which fragments the industry into disjoint vertical markets, limiting the overall market growth. Standardization enables interoperability, which can add value to existing technologies; offer choices and lower costs to technology consumers; and create a horizontal market for technology providers in addition to vertical ones. It is a win-win for everyone. In essence, a standard can bring “discipline” to a fragmented industry, steering investment and innovation towards a common direction to create leverage, thereby increasing the overall efficiency of the industry. Indeed, standardization created an inflection point in the history of many industries.

The content management (CM) industry is no different. The success of the CM industry in recent years has created “content silos” in enterprises. The result is difficulty for an enterprise to reuse or integrate contents that are trapped in disparate repositories. Furthermore, it is expensive for an application developer to address multiple repository markets. The remedy for this “silo” problem is an interoperability standard for CM. “Content silo” is not the only problem the industry is encountering. Driven by Web 2.0, REST, mobile application, and social computing, a technology trend (and thus user expectation) has emerged in recent years for on-demand access to information. The traditional method of accessing business content, using pre-written, repository-specific code, does not allow a user to access content sources discovered at run-time. In contrast, a generic client that uses a standardized interface will be able to access any repository that supports the standard. Going forward, the IT world is moving towards Cloud Computing. In this new paradigm, physical information sources will disappear into the Cloud, and the Cloud will become a big virtual repository with no rigid boundary. How would a client access this virtual repository? An interface for this virtual repository has to accommodate different information sources, including ones that may not have been created yet. An interoperability standard can certainly play a major role here.

Yet, a standard by itself is not always a silver bullet. It is merely a means to solve a problem, not an end in itself. History has showed that not all computing standards have lived to their expectation. To be effective, an interoperability standard needs to be widely adopted by the industry that it is targeted for, both on the consumer side and on the provider side. Developing a specification and getting it ratified as a standard is only half of the task. The other half is to get it adopted by the industry. For CM, these two challenges are not separable in my opinion. One cannot just create a standard and expect vendors to drink the Kool-Aid. For this reason, the CMIS team made an effort to address the adoption challenge

from the very beginning, from engaging stakeholders, to defining the Technical Committee charter, to the design of the data model and interface, and to conducting plugfests to test interoperability.

Adoption is a big challenge for the CM industry, because it cannot afford to ignore the past. In the case of the DBMS industry, without very many widely deployed DBMS's at the time, the SQL standard had the luxury of creating a new RDBMS industry afresh. Back then, the main-stream DBMS technology was IMS from IBM, who invented SQL and decided that RDBMS would be a separate technology and business from IMS. The CM industry today does not have such luck. Over the years, the industry has invested billions of dollars by enterprises, application developers, and repository vendors, accumulating enormous amount of business contents and many applications in support of on-going business operations. There is little chance that the CM industry could start afresh. Migrating content into a new repository and re-implementing all the applications is not an acceptable alternative. Hence, protecting past investments is a foremost goal for CMIS. Until CMIS achieved a reasonable level of adoption, its core mission is, humbly, to find a common ground among the stakeholders. The goal is not to seek technological perfection, but to find a workable solution. CMIS makes no lofty attempt to invent an ideal system and prescribe what a repository "should be".

Adoption must cover both consumers and providers. At this early stage, CMIS consumers are the beneficiary of the standard whereas CMIS providers have to absorb the added costs in order to play. For CMIS v1.0, we focused on a few use cases that offer a reasonable value to the consumers while keeping the barrier of adoption low for providers. For CMIS to establish its foothold, provider adoption is essential because it creates value for interoperability. Without providers, there will be nothing to interoperate with. Limiting the functional scope at this early stage is only a jump-start measure. A more fundamental consideration is to protect past investments as mentioned above. This has to start from the provider side, and it touches upon many design issues. A key design objective for CMIS is that it must be easily layer-able on top of a repository's native interface for most repositories. If a provider has to change the capability or behavior of a repository server in order to comply, the cost will be very high, and the change will likely break existing applications as well. In contrast, implementing CMIS services on top of a repository's native interface allows all existing applications to function normally. In this way, both provider's and consumer's investments are protected. CMIS merely provides an additional, interoperable interface to an existing CM system for accessing its live data.

Achieving this objective is not trivial. There are many subtleties in the data model and interface design that are results of this objective. For example, CMIS defines four kinds of objects as four separate root types, representing four kinds of interoperable CM entities. Were we designing a data model for a standalone CM system instead, a more coherent and elegant design would probably be a single type hierarchy with these four types being extensions, in a certain manner, of a common root type. If CMIS followed this approach, it would impose a specific relationship among these four types, which offers little value from an interoperability perspective. Yet, a system that happens to support a different relationship among these four types will have great difficulty to comply. Let us say, a system's native folder type does not appear on the type hierarchy in the manner prescribed by the standard. Then, for the sake of compliance, a separate "CMIS-folder" type could be created in the proper place on the type hierarchy to support CMIS folders, and CMIS folder services have to be implemented for this new object

type. This amounts to building another folder system separated from the system's native folder system, which is an expensive proposition. Furthermore, all the native folder objects in the system would not be interoperable with CMIS-folder objects. In other words, achieving functional compliance, even at a great cost, does not lead to data interoperability. CMIS's separate-root-types design offers maximum flexibility for mapping separate interoperable concepts to disparate implementations. This example also illustrates that designing an interoperable interface is a rather different challenge than designing the interface of a standalone system. Very frequently, less is more.

If CMIS data model and services are to be mapped to almost all repositories, one might think CMIS's capability would be ultimately bounded by a "common denominator" of all repositories. Not so. There are a variety of features that are important for interoperation but are not supported by all repositories. They are supported by CMIS as optional features. These optional features are discoverable by a consumer programmatically. Vendor support of these features is left to market dynamics, not by mandate. At the other extreme, CMIS is not aimed to cover all the capabilities of a full-function repository either. Technically it is not possible to create a specification that covers all the capabilities of all repositories, simply because there are mutually exclusive capabilities between vendor repositories. Even if this is possible, no provider would be able to comply with such a monster standard. This leads to the hard reality that CMIS should not be expected to replace the native interface of a full-function repository, even though CMIS functionality will undoubtedly expand over time and gradually become "more complete". In fact, having CMIS as "the" interface is not a blessing for a "leading-edge" repository. A native interface is where innovation, technology advancement, proprietary features, and up-the-food-chain venture can be exposed without being constrained by a standard. After all, CMIS is a vehicle to break down content silos. It should not itself create a silo for technology.

One might wonder, would CMIS be forever biased towards providers and handcuffed by legacy technology? Absolutely not! Once provider adoption reached a certain level, the cost of supporting "the next feature" will decrease. As demand for new capability increases, priority will gradually shift to consumers. In the end, technology should be driven by need. A standard that inhibits technology advancement or impedes value delivery simply will not last.

A great deal of hard work, cooperation, and patience from members of the widely participated CMIS Technical Committee went into CMIS v1.0. But the journey has just begun – a journey that takes on the challenge of protecting past investments, the challenge of mitigating the present "content silo" problem, and the challenge of preparing for Cloud Computing of the imminent future. This endeavor hinges on our ability to find a common ground among all the stakeholders. It goes beyond the Committee. It relies on a support by the CM industry at large. The CMIS Products Guide compiled by AIIM contributes to this endeavor, so is each product listed in this Guide. Any effort by readers of this Guide to try out CMIS technology does so too. On this note, I hereby thank you in advance for your contribution to this endeavor.

July 2011