In my first “Standards Now” column, “Defining Our Terms,” I put forward the modest idea that any standard could be defined at its most fundamental level as “anything agreed upon by more than one party.” Later in that column I promised to explain the role of national body representatives for formal standards developing organizations (SDOs).

This topic and the associated range of definitions for what constitutes a standard is the tip of an iceberg of related definitions that can cause discussions on the value of standards to founder. Conventions such as APIs, protocols, and methods constitute the prominent tip of this iceberg of definitions. Hidden under the surface is a variety of organizations of many shapes, types, and sizes that at their core comprise communities of one sort or another. This column aims to explain the role of these communities in developing cloud standards.

The Role of Communities in Developing Cloud Standards

In the world of cloud computing, we’re in the midst of an era in which successful deployments of new applications, interfaces, and any associated programming methods or protocols are driven by simultaneous needs for discoverability, flexibility, and practical usability. The success of new approaches is determined almost immediately by their ability to adapt to such needs in an environment of rapid change.

The Obsolete Terminology of “De Facto” vs. “De Jure”

A range of misunderstandings can arise around the use of the now nearly obsolete terms “de facto” and “de jure” with respect to standards. These terms are generally used to divide extremes of the development spectrum, but they’re almost equally as often distorted and misunderstood. The “de facto” label is commonly attached to something that is simply popular, as opposed to “de jure,” which means literally “of the law” but is sometimes applied to any formal specification.

Unfortunately, the application of this terminology is often subjective. Depending on their orientation, the person or organization using these terms might deploy either label, de jure or de facto, to support arguments for a preferred approach. The latter is often used as a sort of trump card to avoid examining details of a design. Even worse, it can be used as a substitute in an argument for simply picking or favoring a particular vendor’s approach.

As a result, few uses of the de facto and de jure terminology that I’ve seen in the recent press have actually been useful. Such uses generally fail to distinguish between design-driven approaches, which can come from both sides of the development community, and those that simply exist at a given moment and are thus subject to disruption based on developments that are coming down the pipeline from efforts that might have been underway for a long time.
Few existing SDOs operate by force of treaty or law, because governments have deliberately eroded the legal basis for such operation in favor of laws that encourage the recognition and adoption of fieldspecific consensus-based standards. As a result, many types of organizations are now working to develop industry-specific standards. This is a good thing, because it opens up the space of standards development to a wide range of industry and community groups and consortia; however, at the same it contributes to complexity, because the space of such efforts becomes more widespread and disparate.

In addition, open source projects sometimes explicitly position themselves as intending to produce or promote the use of community-based standards. Examples in the non-cloud space include OpenOffice.org in promoting the Open Document Format. Examples in the cloud space are abundant, although spotty in terms of their effectiveness in identifying and promoting widely accepted cloud standards.

What people typically mean when they use these terms today would be far better matched by translating them more loosely as “existing” versus “by design” instead of their literal Latin or historical meanings. As I mentioned earlier, the legal basis for designating some standards organizations as de jure has largely been lost, and in any case currently has little effect on the practical development of cloud standards. Using a looser definition lets us reduce the conflict between these categorizations considerably.

Existing software often incorporates definite features by design, and preselected designs can and in general should exist in any forward-looking product or project. Thus, the distinction between by design and existing isn’t absolute, and there’s room in modern software design for both approaches. The point of standards development should be to produce specifications that promote interoperability between products, or that lay the basis for regulation that could, for example, protect the interests, rights, and privacy of consumers.

So let me be clear: the terms “de facto” and “de jure” are obsolete, and no longer relate to how either software or standards are being developed in cloud computing. Let’s stop using these terms. Instead, we should talk about the level of formality of specifications, and their integration into the software design process.

### Approaches That Work in Practice

The unifying point of this discussion is that both current ultrafast modern software development practice and the effective development of new usable industry standards are driven simultaneously by identifying approaches that will actually work in practice. Each of these development approaches will ultimately meet the other with results that will succeed or fail on the field of user experience. This fate binds software and standards development together at a fundamental level.

Many software developers trying out new ideas for the first time wish to implement their ideas as fast as possible, so seek approaches that will yield new results in the shortest amount of time. People and organizations that develop standards also try to anticipate the directions of new approaches, and steer their efforts toward identifying patterns that will work in the field to bring coherence, interoperability, and other desirable design features to these new efforts. Sometimes these motivations are driven by hard experience from a previous attempt. Each of these approaches is a different but intrinsically related form of development.

What works best in practice is to engage each portion of the community actively to synchronize these different forms, with each anticipating and reinforcing and ultimately influencing the results of the other. That ideal is sometimes missed, and within this point lies the need for outreach on the part of SDOs, an equivalent need for engagement in SDO processes by software developers, and a continuing need for development of community-based methods to bridge the gap between theory and implementation.

Someone approaching interface design from a theoretical line of thought might be swayed by such design characteristics as adherence to a design principle or place within a similar suite of standards. A practical-minded person who’s only interested in what can be made to work right now and can be bought or installed this week, might be tempted to discard such design considerations in the belief that what appears to be the dominant software at the moment has won some sort of Darwinian survival test, and is thus entitled to “set the standard.” This approach might leave the naïve implementer blind to the arrival or availability of a better approach that could be just about to enter the market.
To further make our way out of the de facto/de jure terminology trap, consider what an SDO usually intends to achieve when it produces and labels a specification as a “standard,” and how this usually differs from the meaning of that word in general use. Let’s also take the opportunity to distinguish among various types of standards organizations in terms of formality of procedure and ing their output to more formal groups. The System for Cross-domain Identity Management (SCIM; http://tools.ietf.org/wg/scim/), for example, started as an email list and is now an Internet Engineering Task Force (IETF) working group with three active specifications in progress. Even single-person specification publication in community settings such as GitHub is possible. Many other
the nature of their interactions with the software development community. Many types of such organizations exist, and collectively they can bridge the gaps between formal SDOs and the development community at large.

Community-Based Standards Development

As I’ve mentioned, a wide variety of organizations develop standards. One of the most exciting current trends is the degree to which public methods of organizing work on a specification with the intent to develop it into a standard can be conducted with very few preliminary conditions.

In the last several years, I’ve seen several efforts that started out as no more than email lists or public social media groups turn into dynamic vehicles for developing true community-based standards. These, of course, have generally not yet turned into products that have the force of law, but some of them have matured to the point of taking such examples exist, including several recent efforts to organize specifications on topics related to API design and discovery, which I’ll talk about more in the next issue.

Some organizations work to produce standards that are sufficiently well-defined and formal to be adopted into regulation by governments. Of course, they aren’t the only organizations to operate on a cross-national or global community basis, but they produce a more formal set of specifications and operate according to specific sets of rules. The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) is the only major SDO in this sector still operating with formal treaty status. Accordingly, delegations consist of representative groups appointed by the government of each participating country.

Other international SDOs include the International Organization for Standardization (ISO), a private nonprofit based in Switzerland that has no treaty status, and the International Electrotechnical Commission (IEC). Other national and regional organizations, such as the American National Standards Institute (ANSI), the European Telecommunication Standards Institute (ETSI), and the Japan Industrial Standard Committee (JISC), have varying relationships with their constituent groups. Specialized standards organizations are sometimes formed to pursue a single set of specifications on a particular topic and often operate with only partial or even no contact with larger-scale SDOs. The latter category of organization developed many of the current cloud standards discussed in last issue’s column.¹

Although the rules for participation vary from organization to organization and from country to country, and few of these organizations operate under any official legal mandate, all have formal rules for participation. This level of formality often frightens away members of the software development community, especially individuals, who might otherwise have something to contribute. It also plays a central role in the antipathy that can, and often has, developed between people who would like to move fast in a certain direction but don’t want to petition their government to be involved in a standards body meeting, or who rightly see the burden of pursuing such participation as prohibitive.

Both open source projects and closed source commercial cloud software companies can benefit from deliberately mixing their communities with those of these standards organizations. In fact, several of the working groups in large-scale international SDOs have recently moved away from requiring national body representation during working group meetings in favor of an “everyone is an expert” participation style. These large-scale SDOs actually function as sets of working groups,
similar to the structure of large-scale software projects.

To get out of the trap with which I opened this column, we must realize that neither end of this spectrum of development could exist without the other. Cloud interfaces require the Internet to work, which requires agreements within organizations such as the IETF. The IEEE’s 802.11 standards, for example, help organize wireless Internet connections worldwide. Another example is the area of network function virtualization (NFV), which is attracting interest from people across this spectrum and is spawning new organizations, such as the Cloud Ethernet Forum (www.cloudethernet.org). Many other examples abound, as I discussed in my first column.1

Dialogue between community- and industry-based standards organizations and the cloud software development community is clearly both needed and healthy. Participating communities should include those of open source projects such as OpenStack, CloudStack, and OpenNebula, developers and users of closed source cloud software products, and members of both informal and formal standards projects. The lack of such communication should be viewed as a failure we should all work to avoid.

A Better Model and the Role of Community Activity

Joint membership is one of the best models for bridging the gaps among the various levels of organizations. In this model, individuals or organizations participate in more than one organization, working to bring the perspectives of one approach or organization to another. At a minimum, joint representation by formal contacts among organizations and between SDOs and project communities can also work.

Definite efforts to improve the tone and variety of this dialogue among developer and standards communities can benefit cloud computing. Joint membership among multiple organizations, or at minimum communication between them, can bridge the divide that would otherwise permanently separate development of industry standards from individual efforts within particular software or hardware projects.

THE PROBLEM TO BE SOLVED

Thus comes into focus as follows. If international, mutually agreed upon standards are to exist in rapidly developing fields such as cloud computing, we need sufficient numbers and varieties of intermediate organizations, from community-driven informal efforts to formal SDOs with the explicit cooperative goal of bridging enough of the intervening communication gaps that would otherwise be fatal to such efforts.

Clearly, this situation will be complex and multidimensional. In fact, this expected situation greatly resembles what we have now in cloud computing. Although there are failures in communication all around, there are also many examples of successes, both current and in progress. As I mentioned earlier, the development and ascendance of APIs to their current central role in cloud computing is a particularly interesting example. My next column will explore this topic in detail.

Reference


ALAN SILL is an adjunct professor of physics and senior scientist at the High Performance Computing Center and directs the US National Science Foundation Center for Cloud and Autonomic Computing at Texas Tech University. He also serves as vice president of standards for the Open Grid Forum and cochair of the US National Institute of Standards and Technology’s “Standards Acceleration to Jumpstart Adoption of Cloud Computing” working group. Sill has a PhD in particle physics from American University. He’s an active member of the Distributed Management Task Force, IEEE, TM Forum, and other cloud standards working groups, and has served either directly or as liaison for the Open Grid Forum on several national and international standards roadmap committees. Contact him at alan.sill@standards-now.org.