# A Proposal for Qualities for Smart Objects

James L. Crowley, Joëlle Coutaz

# A Proposal for Qualities for Smart Objects

**James L. Crowley**
Université Grenoble Alpes
Laboratoire Informatique de Grenoble
INRIA Grenoble Rhônes Alpes Research Center
James.Crowley@inria.fr

**Joelle Coutaz**
Université Grenoble Alpes
Laboratoire Informatique de Grenoble
Joelle.Coutaz@imag.fr

**ABSTRACT**
In this position paper, we explore qualities for describing the autonomy, suitability and composability of smart objects. The specific set of qualities proposed below is tentative and incomplete. Our intention is to trigger discussion and debate within the scientific community in order to arrive at a consensus. If successful, a longer, more complete paper will then be prepared to communicate the results.

**Author Keywords**
Software Qualities, Smart Objects, Situated Interaction

**ACM Classification Keywords**
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

## SMART OBJECTS

Smart objects are ordinary objects that have been augmented with computation and communication as well as abilities for perception, action and/or interaction. Smart objects can work individually or collectively to provide services for users where services can be defined as actions and/or information that provide value. As the field of smart objects matures into a recognized engineering discipline, there is an increasing need for tools to model and predict the properties of smart objects. Qualities provide a principled approach to define metrics for performance and are essential for a scientific approach to the development of smart objects.

## QUALITIES

Qualities are characteristics that can be used as normative references for the development process of products, from requirements specification to evaluation, as well as for products comparison.

As global trade has matured, the need for international agreements on quality assessment has become important. It is not surprising then that we are facing the existence of a number of quality models from distinct areas. What these models have in common is the hierarchical organization of quality characteristics into sub-characteristics where the leaves consist of quality attributes that can be measured.

Many terminologies and concepts are used for discussing qualities in different fields, and it is not uncommon to find different terms for similar concepts or even to find the same term used with different meanings within different communities. It is also not uncommon to find terms and concepts at different levels in a hierarchical organization, depending on the nature of the objects studied within different communities. What is most important is that members within a community reach agreement on the concepts and terms for quality as well as their hierarchical relations.

Smart objects are hybrid systems composed of physical and software elements. Thus it is natural to use qualities from both manufacturing and software engineering as inspiration for qualities for smart objects. Garvin proposed eight critical dimensions of quality that serve as a framework for analysis in manufacturing: performance, features, reliability, conformance, durability, serviceability, aesthetics, and perceived quality [1].

Since the seminal work of McCall [2] and Boehm [3], the software engineering community has developed a variety of quality models until some consensus was reached with the ISO/IEC 9126 proposition [4]. To reflect the different stages in the software development process, a distinction is made between quality in use, external quality and internal quality. *Quality in use* is "the capability of the software product to enable specified users to achieve specified goals with *effectiveness*, *productivity*, *safety* and *satisfaction* in specified contexts of use".

Quality in use contributes to the specification of the requirements for (and depends on) external quality. *External quality* is "the quality when the software end product is executed, which is typically measured and evaluated while testing in a simulated environment with simulated data". In turn, external quality defines the requirements for (and depends on) the *internal quality* (i.e. the developer's view of the system under development). External quality is structured into six characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. These categories are refined into sub-categories. For example, usability covers understandability, learnability, operability, and attractiveness.



IUI 2016 Workshop: Interacting with Smart Objects, March 10th, 2016, Sonoma, CA, USA

In Human Computer Interaction, The IFIP WG 2.7(13.4) on Engineering HCI has concentrated on the understandability and operability aspects of usability, but with the use of different terms [5]: *interaction flexibility* to refer to "the multiplicity of ways the user and the system exchange information during task execution" and *interaction robustness* to denote "the capacity of the system to support users in achieving their goals successfully". These subcategories are in turn refined into measurable properties.

All these early (and ongoing) works on quality models and quality assessment are motivated by rationalism, utilitarism and immediacy. There is very little room left to phenomena that develop over time such as attachment, or to characteristics of products that exceed expectation, nourish dreams, or simply human values.

Based on these early works, we propose a three-level hierarchy of qualities. At the top level, we promote four families of qualities that concern Autonomy, Suitability, Durability and Composability. Within these families, smart objects can be described by intrinsic and extrinsic qualities.

Intrinsic qualities [6] are properties that describe a smart object, independent of any interaction with its operating environment. Intrinsic qualities include such characteristics as size, weight, and power requirements as well as properties such as reliability, availability and security. In most cases, intrinsic qualities reflect how well a system or object complies with functional requirements. These can be used to define performance metrics that can be tested under controlled laboratory conditions.

Extrinsic qualities describe how the object or system interacts with its external environment, including users. We include durability, usability, controllability and trustworthiness. Measurement of extrinsic qualities generally requires deployment and evaluation under real world conditions.

Time emerges as an interesting base for organization. In general, for most qualities, there is a basic static definition, and a set of possible projections over time. When possible, we will follow this model for defining qualities.

## AUTONOMY

Autonomy is the ability of a system to maintain its own integrity. For a smart object, autonomy depends on a number of component qualities including energy management, reliability, and durability. Thus autonomy is a composite quality whose definition and measurement depends on it components.

Failure, for a smart object, is the partial or total loss of system availability. Failure can be temporary or permanent and can refer to specific functionality or overall system operation. Many different quality metrics refer to conditions of failure.

### Power autonomy

Systems require energy and consume power. Energy is the capacity of a system to perform work, measured in Joules. Power is a measure of the energy consumed per unit time, measured in Joules/Second. Average power consumption, as well as minimum and maximum instantaneous power requirements are important qualities for any system.

Smart objects typically rely on electrical power for energy. Electrical power is measured in Watts defined as Joules/Second. Minimum and maximum power requirements of a smart object are measured in Watts, while average power consumption is measured in Watt-Hours. Standard tools and techniques exist for measuring electrical power consumption, and these can and should be adopted for measuring and monitoring power consumption.

Power autonomy is the ability to operate without a physical connection to an external power supply. The need for power autonomy frequently raises important challenges in the design of smart objects. In addition to instantaneous and average power requirements, the duration of power autonomy can be an important factor in the commercial value of smart objects.

### Reliability

Reliability is the ability of a system to consistently perform its required function without degradation or failure over time.

Among the most common measures of reliability are the mean time to first failure, the mean time between failures (MTBF) and the failure rate per unit time [1] where failure can be a partial or total loss of correct function or even a loss of availability.

Availability can be an important component of reliability. Availability measures "the proportion of total time during which the system is an up state" [2], that is how often the object is available for use, even though it may not be functioning correctly. Availability can enable recovery through external intervention.

More detailed measures of reliability depend on specific functions. These can include probability of error, and failure rate.

### Durability

Durability is the ability to withstand repeated use over a period of time without significant deterioration in performance. Durability is distinct from reliability in that it includes the way the object is used as well as the time over which it operates [1]. Durability can be defined as the amount of use one gets from an object before it fails while reliability is the length of time between failures regardless of use. For example, the number of recharge cycles of a lithium ion battery is a measure of its durability, rather than its reliability.

For smart objects, two notions of durability are relevant. The first, obvious, notion refers to the device life cycle (longevity). This can be measured by the number of hours of operation or by the number of times a device can be turned on and off.

A more interesting notion of durability captures the ease with which a device resists obsolescence. Causes for obsolescence are three-fold: software, material, and people. For the software components of a smart object, durability can be measured in terms of maintainability and portability. For the material/hardware components, durability can be measured with the lifespan of its materiality, but also by its capacity to be re-cycled and up-cycled, along with its environmental impact from the extraction of raw material to disposal.

People discard some objects, although they are still functioning, while they preserve others because of some form of bonding that builds over time. Early studies show that factors that affect attachment include [7]: engagement ("the extent to which an object invites and promotes physical engagement with its owner during use"), histories ("the extent to which the object preserves personal memories"), and augmentation ("the extent to which an object can be reused, modified, altered beyond its original use". Composability, which permits incremental changes, is thus an important characteristics to consider for smart objects.

## COMPOSABILITY

Composability refers to the ability of the object to function as part of an assembly to collectively provide a set of services that makes sense for users. Objects can be assembled mechanically, electrically, wirelessly or functionally. The essential features that make a smart object amenable to composition are the generality, diversity, cardinality and difficulty of its interconnections.

Generality, diversity, cardinality and difficulty can be illustrated using electrical interconnection. An object can be characterized by the number of different types of physical connectors (diversity), the number of different types of other objects that can be connected via its physical connectors (generality), and the total number of physical connectors available (cardinality). Similarly, for wireless interconnection, one can describe the generality of the wireless protocols available, the diversity of the types of wireless protocols, and the cardinality of connections that can be maintained at any one time.

Functional composition can be formalized in terms of software services [8]. An object can be characterized by the overall number of composite services that can make use of its functions (generality), the variety of different kinds of composite services to which it can contribute (diversity), and the number of composite services to which it can contribute at the same time (cardinality). Pushing the analysis further, a distinction can be made between syntactic composability (e.g., are data types and parameter passing compatible?), and semantic composability (e.g., is data from the source service within bounds expected by the sink service?).

Additional aspects of composability concern property preservation, temporal aspects of composition, and the amount of effort required to establish an interconnection:

- Does the assembly preserve the properties carried by the smart objects individually? For example, if two smart objects satisfy observability, is observability still supported by the composition (in particular, is the state of the interconnexion observable)?
- Is the composition - reconfiguration, and decomposition, static or dynamic? In turn, dynamicity supposes the existence of the appropriate underlying middleware whose functioning should be amenable to people.
- Is the assembly performed automatically by the objects themselves, manually by users, or by the cooperation of both? In case of human intervention, what are the cognitive-sensory-motor efforts required? What is the most suitable approach?

## SUITABILITY

Suitability is the capacity to act and interact in a manner that is appropriate for the task and context. Suitability can be characterized by a variety of properties ranging from intelligence to usability, controllability, aesthetics and trust.

### Intelligence

Simply augmenting an object with micro-electronics does not make it smart. To be smart, an object should behave in a manner that is appropriate for its role and its environment. Indeed, in many cases the most basic quality for a smart object is smartness, defined as the appropriateness, or intelligence, of its behaviors as judged by an external observer. In robotics, appropriate behavior is commonly referred to as situated [9].

### Usability

As discussed above, a number of quality models, factors and criteria have been developed to define and assess the usability of interactive systems from desktop computers, tablets, tabletops, smart phones, and even cars and mobile robots. All of these are valid for assessing the usability of smart objects, although the notion of "useworthiness" is not sufficiently addressed.

Useworthiness is central to Cockton's argument for the development of systems that have value in the real world [10]. In value-centered approaches, design starts with an explicit expression of the potential values for a set of target contexts. Intended value for target contexts are then translated into evaluation criteria. Evaluation criteria are not necessarily elicited from generic intrinsic features such as time for task completion, but are contextualized. They are monitored and measured in real usage to assess the achieved value.

### Controllability

For smart objects, controllability is the ability to regulate, dominate or command the behaviors (action and interactions) of the object. Control of ones' personal environment is an important component of general well being, and can be an important factor in the rate with which individuals will invest time and money in smart objects and smart services.

Loss of control (or preemption) is an important aspect to measuring controllability. Thus, controllability can be measured in terms of the number of behaviors that can or cannot be selected at any time,

### Privacy and Security

Privacy is the ability to protect information from disclosure or observation. Security is the ability to assure that both information and system components have not been subject to unauthorized modification or disclosure. Privacy and security are closely related but separate properties.

Note that security is distinct from trust. Security refers to the ability of the system to withstand attack, while trust refers to the confidence that users have in the ability of the system to withstand attack. As system may be secure but untrusted, or it may be trusted but insecure.

As a form of protection, both privacy and security are measured as absence of violations. For privacy, this means explicitly listing all information that is or can be disclosed. For security, this means listing the category of attacks that the system is certified to resist.

### Trustability

Trustability is ability to inspire confidence in one or more qualities. This can range confidence in the reliability of the object, in the intelligence of the object or in its privacy and security. As mentioned above, trust in the quality of an object is separate from the quality itself. Because trust is an ability to inspire users, the obvious manner to measure trust is to gather statistics on the beliefs of users with regard to different qualities.

## CONCLUSIONS

In this short position paper we have explored definitions for qualities that describe the Autonomy, Suitability and Composability of smart objects. For each family, we have defined intrinsic qualities that concern innate properties of a smart object as extrinsic properties that describe the interaction between the object and its environment. We have noted that many qualities have a core static definition as well a set of possible temporal projections.

The definitions presented above are tentative and incomplete. Our purpose is to trigger discussion and debate in order to arrive at a true consensus within the scientific community. If successful, the community will have made an important advance towards establishment of the study of Smart Objects as a scientific discipline.

Many of the desired qualities of smart objects arise from a goal of improving quality of life. Quality of life (QoL) is commonly defined as the general well-being of individuals and societies. Metrics for quality of life have been proposed for fields such as healthcare and gerontology. These can be used to define qualities for smart object based on the requirements of target application domains.

The ultimate quality for any system is value. The obvious measure of value is how much wealth a user is willing to pay to assure access to the system. However, there are aspects to value that in some cases defy monetary estimates. Measuring value requires understanding the services that a system provides to a user, and how much sacrifice the user is willing to undergo to assure access to the service.

## ACKNOWLEDGMENTS

This work and ideas reported in this paper have been partially sponsored by the French Agence Nationale de la Recherche (ANR), program “Investissement d'Avenir” project reference ANR-11-EQPX-0002, Amiqual4Home and European program CATRENE project AppsGate (CA110).

## REFERENCES

1. D. A. Garvin, Competing on the Eight Dimensions of Quality, Harvard Business Review, November 1987.
2. J. A. McCall, P. K. Richards and G. F. Walters, Factors in software quality. General Electric, National Technical Information Service, 1977.
3. B. Boehm, J. R. Brown, H Kaspar, M. Lipow, G. J. MacLeod, M. J. Merritt, Characteristics of Software Quality, North-Holland, 1978.
4. ISO/IEC 9126-1:2001, “Software engineering – product quality- Part1: Quality Model, 2001.
5. C. Gram, G. Cockton, “Design Principles for Interactive Software”, Chapman & Hall, 1996.
6. B. B. Agarwal, S. P. Tayal, M. Gupta, Software Engineering and Testing, Jones and Bartlett, 1975.
7. M. Odom, J. Pierce, E. Stolterman, and E. Blevis, “Understanding Why We Preserve Some Things and Discard Others in the Context of Interaction Design”, Proc. CHI'09, ACM, pp. 1053-1062.
8. J. L. Crowley, J. Coutaz, “An Ecological View of Smart Home Technologies”, 2015 European Conference on Ambient Intelligence, Athens, Greece, Nov. 2015.
9. C. Breazeal, Designing Sociable Robots, MIT Press, Cambridge MA, 2002.
10. G. Cockton, “A Development Framework for Value-Centred Design”, CHI'05 extended abstracts on Human factors in computing systems, pp. 1292-1295, ACM, 2005.