Adaptable Interfaces [1]

Ken Krechmer
Lecturer, University of Colorado
Boulder, CO, USA
e-mail: krechmer@csrstds.com

Abstract—**Adaptable interfaces offer the possibility of more maintainable and reliable systems. This paper defines the four ways interfaces can change and develops the concept of adaptability using communications theory. Adaptability is a complex process, which requires a number of supporting processes. Together these process automatically negotiate possible capabilities across an interface. The concept of state-pairs is used to define communicating entities, interface, comparison, measured information, communications, flexibility and finally adaptability.**

Keywords-adaptable; interface; communications structure; measured information; etiquette

I. INTRODUCTION

As systems become programmable at lower layers of the OSI model, the interfaces in such systems can become more versatile. Prior to programmable systems, interfaces were known to be fixed or at most modifiable using an external adaptor. Programmable systems change this interface paradigm. This paper defines the four ways interfaces can change and develops the concept of adaptability using communications theory. Adaptability is a complex process, which requires a number of supporting processes. Together these process automatically negotiate possible capabilities across an interface. Adaptability requires communications across an interface between at least two entities. First the structure of a communications system is developed. From this structure all the processes supporting adaptability are derived. With a more complete understanding of these processes, a more rigorous understanding of adaptability, and the advantages of adaptable interfaces, emerges.

II. A COMMUNICATIONS STRUCTURE

Fig. 1 models communications for the purpose of understanding its structure rather than its performance. Fig. 1 is similar to the Shannon model of a communications system [2] except that the communications channel is replaced by an interface and the probability of the output message being the same as the input message is fixed to one. The transmitter (T) and receiver (R) are independent communicating entities connected via an interface. The purpose of this model is to analyze the structure of the relationship between T and R.

From communications theory, T and R support all the state-pairs t_i - r_i, where i represents the set of all t or r states 1 to n in Fig. 1. A state-pair includes a specific input part (t_x) associated with T, which is related to the output part (r_x) associated with R. An interface describes the one-one relationships between the related parts of two or more state-pairs. "A relation is said to be one-one when, if t has the relation in question to r, no other term t' has the same relation to r, and t does not have the same relation to any other term r' other than r" [3]. All the state-pairs associated with T and R form the interface between T and R. A single set of t_i or r_i states is usually considered a specific parameter (e.g., data rate) of the transmitter or receiver. Communications (information transfer) is possible only when multiple state-pairs form an interface between independent entities. An interface does not exist independently, it is formed by the common parameters of the communicating entities. Most interfaces include multiple parameters.

A one-one relation is in some way a relationship between equal elements. As example in Fig. 1, state t_x may be seen as the equal of state r_x. However, it is not possible to define such equality without specifying other sets of state-pairs. For state t_x to be equal to r_x for example, the boundary conditions, tolerances, and measurement apparatus all must be equal. Such equality is possible in theory but difficult in practice. In practice, the relationship between each state t_x and r_x is more easily described as one-one.

The concept of state-pairs may be applied to any interface, even a physical interface. Examine a perfectly compatible (zero tolerance) physical plug and socket. The outside diameter of the plug and the inside diameter of the socket are the same. The length of the plug and the socket are the same. The physical interface between the plug and socket consists of all the common pairs of points on the plug and socket. These common points are the state-pairs, which form a physically compatible interface. Even in this simple physical interface, multiple layers of sets of state-pairs are needed to completely define the interface. Other parameters that need to be defined include: the physical dimension system used, the tolerances applied, even concepts such as diameter and length have to be "agreed" at each communicating entity.

III. COMMUNICATIONS PROCESS

The ability to pass information across state-pairs requires two comparisons. Each comparison is associated with a part of a state-pair. The fundamental nature of these comparisons is suggested by I. Kant who states that a comparison is necessary for understanding [4].

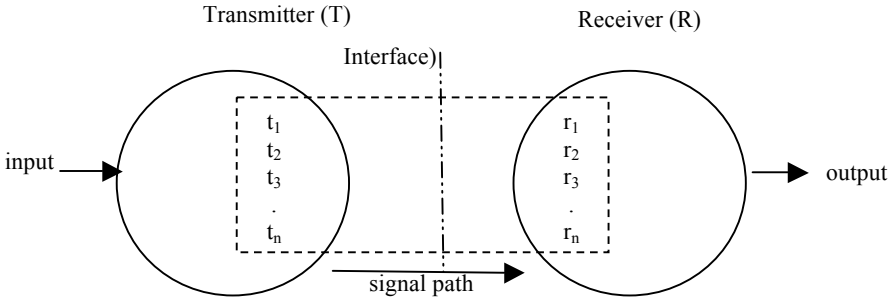

Figure 1. Communications structure.

The simplest communications process may be six operations, three in the transmitter and three in the receiver (Table 1). Operations 2 and 5 in Table 1 demonstrate how a state-pair relates to the communications process. The number of operations is not critical to this analysis. What is critical is that the communications process consists of symmetric transmit and receive processes, each of which includes a comparison. This symmetry of a communications process also appears in the Venn diagram in Fig. 3, below.

Consider a binary amplitude-modulated communications system with two state-pairs (t_1 - r_1 and t_2 - r_2) and without time domain or tolerance effects. The input message to T is compared with the decision boundary between t_1 and t_2 determining which state causes a T signal output. T encodes +V signals for t_1 and -V signals for t_2, which are received as signals in R. The received signal is compared with the decision boundary between r_1 and r_2 determining which state causes a R output message. The decision boundaries, both threshold and maximum, are lower level parameters (formed by other parameters created in the implementation of T and R). These boundaries implement the relationship between each part of the t_1 - r_1 and t_2 - r_2 state-pairs and determine the operational characteristics of the signal path. A more complex communications system has more sets of transmitter and receiver state-pairs (parameters) and more complex boundaries.

Example: In the course of reading, a word appears of unknown meaning. The reader refers to a dictionary. A dictionary relates words (states) to their meanings (message). The author and reader select words from similar dictionaries (first and second comparisons). The author's and reader's dictionaries together are the state-pairs of equal words with a common meaning in each dictionary.

TABLE I. COMMUNICATIONS PROCESS

For the transmitter (T):	
1	Select an input message
2	Compare this input and determine state (t_i)
3	Output a signal
For the receiver (R):	
4	Select the signal received
5	Compare this signal and determine related state (r_i)
6	Output message

The state-pairs in a communications system may be created by chemical bonds (A-C, G-T in DNA), pre-existing written or spoken alphabets, pre-existing dictionaries or syntax, the specifications or standards defining a transmitter, receiver or protocol (electronic communications) or a physical implementation of a transmitter, transmission link, or receiver. Different forms of state-pairs are divided into layers in the Open System Interconnect model (OSI) where each reference layer provides the interface(s) used by the next layer. Layer one includes physical aspects of the interface and higher layers include successively more abstract functionality. All the state-pairs used for a specific functional relationship between two or more entities create an interface.

A. Interfaces

There are four broad catagories that describe how the state-pairs that define an interface may change - fixed (state-pairs are unchangeable), flexible (state-pairs changed by external action), adaptable (state-pairs are selected by negotiation across the interface) and evolutionary (state-pairs change by internal action). These four interface variations are not mutually exclusive and may exist in different combinations at different layers of the OSI model. A mechanical plug and socket is an example of a fixed interface. Using adapters to convert AC power connectors to different countries' power outlets is an example of a way to implement flexibility. Adaptable interfaces are necessary for true peer-to-peer operation. An Internet Engineering Task Force (IETF) protocol Session Initiation Protocol (SIP) used to negotiate capabilities between two communicating ends is an example of an adaptable protocol. An example of an evolutionary interface would be a system that autonomously accesses independent web sites to acquire additional interface capabilities. Enabling automatic upgrades of personal computer software is an example of evolutionary software. An evolutionary interface is yet more complex, as the independent software defining each side of the interface must be upgraded.

Other examples of flexible interfaces include: an Edison light bulb socket that supports many different types of lamps. While the mechanical aspects of the light bulb and socket are fixed, the load can be changed. A human user manually identifies and selects the specific lamp and the

Edison light bulb plug/socket (the physical interface) makes this flexibility possible. A protocol example of a fixed interface that supports flexibility is the use of the Internet protocols TCP/IP as the interface with which each lower physical network or higher layer protocol is designed to interoperate. XML (eXtensible Markup Language) is an example of an interface protocol that supports flexibility and can reduce fixed state-pairs.

IV. MEASUREMENT PROCESS

After defining communications and interfaces, next an understanding of the measurement process is required to understand adaptability. A measurement is a quantified selection of an observable. The process of making a quantified selection is similar to the transmitter or receiver process shown in Table 1 (select signal, compare signal and determine state, output signal). In a measurement process there is a measurement apparatus that compares an observable with a predefined set of states. The states of the measurement apparatus must be related to the observable for a measurement to be practical. A measurement and a communications transmitter or receiver, as described above, are quite similar. This similarity supports the use of communication theory to analyze the measurement process. With a measurement process understood, communicating entities may be defined. Then adaptability can be defined for communicating entities.

N. Campbell defines a measurement (the concept) as "the assignment of numerals to represent properties" [5]. A measurement process assigns the numerals by utilizing one or more comparisons with states of the measurement apparatus. Each of these states of the measurement apparatus, and its associated boundary conditions, acts to quantify the measurement. Any observable that may be quantized, e.g., weight, length, color, hardness, texture, transfer rate, capacity, spin, etc., may be measured. The observable defines the property to be measured and the range of states of the property in the receiver quantifies the measured parameter.

The choice of the receiver states and boundary conditions actually selects the parameter and quantification of the entity to be measured. That is, if a length scale is used, distance is measured; if a weight scale is used, weight is measured; if a voltmeter is used, voltage is measured, etc. A measurement is not absolute; it is always relative to the parameter measured by the receiver, the states of that parameter in the receiver and the boundary conditions between states. A measurement requires that the states of the receiver be represented in a definition of measured information.

Equation (1) is Shannon's equation for entropy [2, page 50]. D (2) is defined in T. Cover and J. Thomas as the entropy relative to *log n* [6, page 27]. This section of the paper develops the theory that D represents the information contained in the measurement of a parameter (T) of an entity A receiver (for t_i) with n discrete states is applied (represented by the first term [*log n*] in (2)). The entropy distribution (H(T)) of the measurement process is calculated by the second term. *p* indicates a probability. The output from the measurement process is one or more specific states t_x, t_y, t_z. This measured information is equal to

$$H(T) = -\sum_{i=1}^{i=n} p(t_i) \log p(t_i) \quad (1)$$

$$D = \log n + \sum_{i=1}^{i=n} p(t_i) \log p(t_i) \quad (2)$$

D.

As example, a voltmeter (used to measure volts) with a 3 volt full scale (parameter of the voltmeter) and the minimum measurable increment (boundary condition) of 0.1 V, has 30 (= n) possible states of v_i and produces a single output measurement v_x, then D = log 30. The greater the number of states n, the greater the information from the measurement process. The narrower the distribution of the entropy term (H(T)), the greater the information. A perfect measurement (zero H(T)) produces the maximum information, *log n*. The first term of (2) effectively includes the concept of tolerance (minimum measurable increment) in the measured information calculation.

Fig. 2 expresses (2) as a Venn diagram. Fig. 2 shows how the limit of the entropy distribution (*log n*) is related to the entropy distribution (H(X)). (3) is Cover and Thomas' equation for Mutual Information (MI), the relative entropy between related entropy distributions. Replacing H(R) in (3) [6, page 19] with *log n* calculates the mutual information of H(T) and *log n* (4).

$$MI = I(T;R) = H(R) - H(R|T) \quad (3)$$

$$MI = \log n - (\log n - H(T)) \quad (4)$$

$$MI = H(T) = I(T; \log n) \quad (5)$$

Equation (5) shows that H(T) when referenced to its limit is equal to the mutual information as the *log n* terms cancel in (4). Thus, using D (2) provides a rigorous description of measured information without changing mutual information (MI).

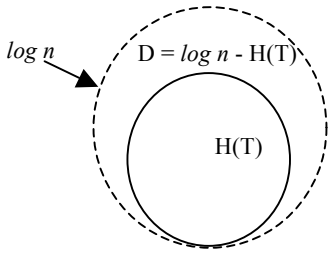

Figure 2. Venn diagram of H(T) and its limit.

A related result to (5) substitutes H(T) for H(R) in (3) and is noted as self-information [5, page 20]. Equation (5) and self-information indicate that the reference may be either *log n* or H(T) itself. If the reference is not *log n* or H(T) itself, then there are additional parameters (not T). A single parameter entropy distribution should be referenced to its limit (i.e., *log n*), as applying H(T) to reference H(T) is self-referential. The measured information related to a single parameter entropy distribution only exists in relation to a reference and the only logical reference is the limit of the entropy distribution. This provides a proof of D (2) as the definition of measured information.

The different observables of an entity are acquired by multiple measurements. The multiple measurements necessary to define an entity may each be represented by a D_i. With a model of an entity, communications between two entities can be defined by relating the Venn diagrams of the transmitter and receiver versions of Fig. 2 in Fig. 3.

V. COMMUNICATIONS

Communications exist when the six operations in Table 1 occur. The comparisons necessary for communications require the existence of common state-pairs between two distinct entities. A communications system may be modeled by using two overlapping Venn diagrams from Fig. 2 as shown in Fig. 3. Fig. 3 is derived from Shannon's model of a communications system, where the receiver output is related to the transmitter input by a probability less than one. In Fig. 3, *log* n_t is the bound of H(T) and *log* n_r is the bound of H(R). The intersection of log n_t and log n_r is shown as a dotted lens shape. This space represents the interface (I) made up of all the state-pairs of T and R. I limits the mutual information (MI, overlap of the solid circles, solid lens shape) possible between the transmitter and the receiver.

Fig. 3 identifies that a communications system creates mutual information by comparing the transmitter and receiver inputs to the respective parts of state-pairs, not by H(T) to H(R) interaction.

Summarizing the communications structure and process developed thus far: Comparisons are necessary for communications and measurement. A measurement, using a comparison, quantifies an element of a set in relation to a reference. A group of measurements defines the observables of an entity. State-pairs form the interface between two compatible entities. An interface allows communications by supporting comparisons between sets of state-pairs Communications between programmable entities can support adaptability. Using this model it is now possible to define adaptability.

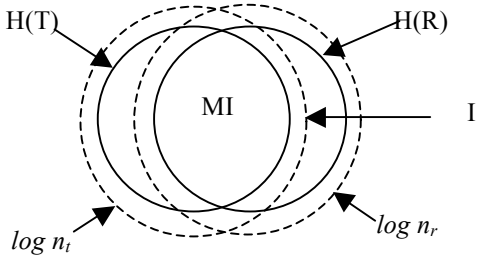

Figure 3. Venn diagram of a communication system.

VI. ADAPTABILITY

Each parameter presented across an interface consists of a number of state-pairs (*n*). However, the number of states in T may not be the same as the number of states in R for some parameters. Such unpaired states occur when parameters, by virtue of options, special features, differing revisions or just non-selection in the transmitter or receiver, are not available or not used. For instance, telephone modems may offer six different modulations ranging from 300 bit/s to near 56 kbit/s. Usually only the 56 kbit/s modulation is in operation and the five other modulations are unused.

Fig. 3 shows unpaired states within each dotted circle areas (*log* n_t and *log* n_r) and outside the dotted lens shape (I). The communications structure is more efficient when unpaired states don't exist. Older communications systems, which tended to be single provider (e.g., telegraph and telephony) tried to avoid unpaired states. Newer communications systems tend to have more and more unpaired states as communications becomes more complex and variation increases. Interconnected systems have become larger, multi-vendor and may include many revision levels and multiple technologies. These increasing trends cause more and more unpaired states.

At least two approaches have been used to avoid unpaired states: 1) the selection of other capabilities has been treated as vendor-specific and not defined (e.g., the 3G cellular IMT-2000 standards); 2) a protocol is defined to determine which of the available capabilities in the T or R should be employed in a specific situation. As example, telephone modems prior to V.32 (circa 1984) selected the modulation to be used based on convention and vendor-specific decision boundaries. After V.32, the identification, negotiation and selection of a specific modulation was defined by an independent protocol, V.8.

The process of automatically negotiating possible capabilities is termed *adaptability* as it makes a system more adaptable. As defined here, adaptability requires three specific functions: identification of the capabilities available at each end, negotiation to determine the desired state-pairs (the interface), and selection of the desired state-pairs (which may require accessing software from elsewhere). These three functions are more complex versions of the basic functions required for any communications: select input, compare input to reference (with adaptability mechanisms each end is compared to the other), and create output (select state-pairs). After these adaptable processes are completed, then information or control communications can begin across the negotiated interface.

Fully flexible interfaces such as XML are the current state-of-the-art. By definition, an XML relationship is not peer-to-peer. Only when the two communicating entities can negotiate any change independently can they be peers.

Adaptability, the means to support such negotiation, may be created by a software program (often termed agent software) that can identify, negotiate and select the state-pairs across an interface. Or an independent communications protocol may be used for the purposes of identification, negotiation and selection. When such a protocol is used only for these purposes, it is termed an *etiquette* [7]. It seems likely that other approaches to implement adaptability may be identified.

Etiquettes are already used in some communications systems, e.g., ITU V.8 for telephone modems, ITU T.30 for G3 fax, ITU G.994.1 for digital subscriber line transceivers, and IETF Session Initiation Protocol (SIP); their properties have been explored previously [7]. In a future 4G cellular architecture, an etiquette could allow the service provider to negotiate the protocol that optimizes system loading or maximizes geographic coverage, or allow a user to select the protocol (and related service provider) that offers the best economic performance for that user. Troubleshooting of incompatibilities using an etiquette is also easier as each end can identify the available and compatible parameter sets of the other end. The use of adaptability mechanisms is a system architecture choice that significantly enhances the long term performance of programmable heterogeneous communications systems.

When systems are programmable, adaptability is possible. An etiquette transmitter presents the range of possible compatible parameters to an etiquette receiver. The etiquette receiver responds with its range of possible compatible parameters. Using heuristics local to the transmitter and receiver (e.g., largest parameter is best [pels, bits, colors, data rate, etc.]) or remote heuristics accessed by both the transmitter and the receiver (e.g., using a remote data base to determine which common parameters are to be utilized), the etiquette transmitter and receiver negotiate and select the desired interface for compatibility and follow-on communications.

Communications interfaces are layered. Adaptability may be employed at each layer of the OSI model or partially in one or more layers. Adaptability could be useful in communications entities such as software defined radios. A software defined radio that includes the physical layer, perhaps others, is not defined as adaptable but has the properties - programmable and a radio interface (non-mechanical) - that allow it to be adaptable. The ability to change the software in a system is sometimes termed reconfigurability or changeability. But these terms do not necessarily denote adaptability. Currently discussions of adaptability do not define which layer or how many layers are adaptable and may confuse the concept of flexibility with adaptability. One purpose of this paper is to better define terms such as "flexible" and "adaptable".

Compatible systems have state-pairs. If there are transmitter states (at any OSI layer) that do not have related receiver states, such inconsistencies cause "bugs". Adaptability mechanisms offer a means to negotiate and select a specific interface and thus reduce such bugs.

For this unique functionality of etiquettes to operate consistently, any addition to an etiquette must be a proper super-set of the previous version. As long as the etiquette is a logical single tree structure, where each branch refers to a single parameter set and no deletions are allowed, a correctly modified etiquette will always be backward compatible. Following this model an etiquette may be expanded whenever desired independently in the transmitter and the receiver. This allows new capabilities, and the parameters in the etiquette that identify them, to be added to a communications system at any time. If both ends can support the new parameters they can be employed. If one end supports a parameter and the other end does not, either this parameter will not be used or it may be practical for the deficient end to download the needed software from a known Internet web site.

Adaptability mechanisms also can support the use and charging for proprietary technology in public standards. If one or more parameters in the logical tree are identified as proprietary (e.g., identified by a trademark), the use of such parameters would legally require the trademark owner's approval. All the other parameters identified in the etiquette remain in the public standard. Such approval might require some form of payment to the trademark owner. If the proprietary technology is valuable, implementers or users will have reason to pay the trademark owner for the use of their proprietary technology. Many different procedures are possible to compensate the trademark owner: charge for downloads, per implementation fees, usage fees, periodic maintenance/support fees, or simply the sales advantages of proprietary implementations offering improved operation over the public sections of the standard.

VII. EVOLVABLE INTERFACES

Looking further into the future: evolvable interfaces have not yet been developed. Such interfaces could enable a new level of system openness. Consider a future open cell phone system where new features may be added to the system by uploading operating software to a specific web site. An independent developer defines and creates software for a new cell phone functionality and the related software for the cell phone system base station. This software is uploaded to a specific web site. When any other cell phone users anywhere finds this capability desirable they could download this new capability to their mobile as well as any necessary base stations either automatically or as desired. Consider what could happen when a developer creates a new video or voice compression algorithm. Using the process described, the new algorithm could be used throughout the cell phone system wherever it was desired. It is also possible to imagine that there is a charging system that allows the developer to charge each user for download or usage of the new algorithm. Then the new capability would be tested in the market to see if users will bear the required charges, rather than being forced on the users by the original system designers' decisions.

VIII. CONCLUSION

Adaptable interfaces makes it possible to automatically negotiate the rising complexity of communications, introduce new technology into communications channels at will, simplify communications troubleshooting, better support multi-mode operation, avoid identified communications channel bugs, and support incentives to developers and implementers without forcing all users of public interfaces to pay private fees. The advantages of making all programmable interfaces adaptability are significant enough to suggest that adaptability should be a requirement for future programmable interfaces in communications systems.

REFERENCES

[1] An earlier version, titled Quantifing Adaptability, was presented at IARIA ADAPTIVE 2010, The Second International Conference on Adaptive and Self-adaptive Systems and Applications, November 25, 2010, Lisbon, Portugal.

[2] C. E. Shannon and W. Weaver, The Mathematical Theory of Communications, Fig. 1 p. 34. Urbana and Chicago IL, USA: University of Illinois Press, 1963.

[3] B. Russell, Introduction to Mathematical Philosophy. New York: Simon and Schuster, 1971, page 15.

[4] I. Kant, Logic (General Doctrine of Elements, Para. 6, Logical Acts of Comparison, Reflection and Abstraction), Library of Liberal Arts, trans. R.S. Hartman and W. Schwarz. Indianapolis and New York: The Bobbs-Merrill Company, Inc., 1974.

[5] N. Campbell, Foundations of Science, p. 267, Dover Publications, New York, NY, 1957.

[6] T. M. Cover and J. A. Thomas, Elements of Information Theory, New York: John Wiley & Sons, Inc., 1991.

[7] K. Krechmer, "Fundamental nature of standards: technical perspective," IEEE Communications Magazine, 38(6), p. 70, June, 2000. Available at http://www.csrstds.com/fundtec.html