Seven Habits of Highly Effective Interface Design

David Broberg, Saturday, © June 23, 2001

One of the most critical design elements when developing new technology is the creation of new interfaces. New technologies such as DTV, PVRs and digital cable require new architecture. New dividing lines are formed when developing the architecture and deciding what functions and features should go in which box. This division of function leads to interface design to enable the flow of signals from one part of the architecture to the other. These interfaces can be internal and may merely allow circuits on one board to talk with those on another board, or they can be external allowing a *source* device to send its signals to the *sink* device.

Below, I present my idea of the seven habits of high effective interface design. This list of habits or principles is not intended to be comprehensive, and probably not all of them can be applied to every situation. They are intended to be helpful guides to developing interfaces that are most effective.

1. Interface design should minimize duplication of components. The first principal says that when possible, all care should be taken in the division of the architecture to avoid duplication of functions. For example, in designing an output of a set-top-box, it is not necessary to include signals that are consumed or digested by the set-top. Since the set-top-box is the primary navigation device, it is not necessary to pass navigation information beyond the set-top-box. Doing so would create user confusion and duplication of circuitry.

2. Interface design should minimize redundant operations or settings. Once again, redundant settings or operations should not exist on both sides of an interface. The easiest example to think of is the volume control. While it is always possible to add a volume control on either side of an audio interface, it is not helpful to do so. Duplicating a user control or setting causes confusion and poor performance. Imagine the audio example where the output-control is turned way down, and the input-control is turned way up. The results will be added noise and distortion. The same is true for video adjustments and other signals.

3. No interface should preempt downstream functionality. An interface that provides compressed signals should include all the signals helpful in the decoding of the signals. To strip away signals from a compressed interface such as aspect ratio tags, or colorimetry descriptors, would prevent the full functionality of the downstream decoding device.

4. Signals passing an interface always should be described accurately. This principle seems simple enough, but in some cases interfaces are defined with a certain choice for colorimetry, aspect ratio, or some other signal parameter with no way for the source device to ensure that the signals passing the interface conform. You can't specify what you need downstream if the upstream source does not have control over the parameter.

5. Excessive or redundant processing should be avoided whenever possible. For example, transcoding and format conversion may be necessary on both sides of an interface, but the need for redundant processing should be minimized by the choice of formats. The architecture should be carefully divided so that all format conversion is primarily on one side of the interface or the other. For compressed or encoded interfaces, the processing should be on the sink side. For uncompressed or decoded signals, the format conversion should be on the source side.

6. Be a good neighbor on the interface. This is really the principle of doing no harm to others. Adequate care should be given in the interface design so that the source device can communicate

any needed warnings effectively to the sink device. One side of the interface should never be designed in such a way that it may cause harm to the other side of the interface. This could be as simple as properly matching source and sink for power signals, or as complex as intelligently avoiding long periods with stationary images which might cause uneven phosphor aging of the display

7. Protect copyrighted material. The final principle for effective interface design is to ensure a mechanism exists for the protection of intellectual property rights of content passing across any external interface. Effective copy protection usually involves some authentication process to ensure that the device connected on the other side can be trusted. Adequate copy protection also relies upon some type of signal scrambling which is used to prevent snooping. This ensures that only authorized devices on the other side of the interface, which protect the content, can be used.

If you participate in the development of interfaces or architecture design for new technology, you can be more effective by following these seven habits. Designs that take these seven principles into account will be more cost effective, and easier to use than those products which neglect even a few.