Sample Projects

HSBC, Video Conference and Presentation Systems
Global Rollout

- HSBC had the largest Cisco VC estate in the world with over 50,000 endpoints in over 10,000 buildings throughout the world.

- The bank required the following:

    - Standardized video conference and presentation system which could be deployed in various room settings and sizes.
    - Standardized program and touchpanel layout.
    - Successfully integrate with the banks TMS server for address book and call merging capabilities using secure SSH.

- An initial demo system was setup on the 9th floor of the HSBC HQ in Canary Wharf, London to allow for extensive development, refinement and testing in coordination with the bank and its key representatives.

- The system contained the following key aspects and functionality which were critical to its success:

    1. Self-Test:

        - Full self-test available from the touchpanel displaying all real-time Tx/Rx with all components, with red/green indicators should any fault be found with any component.
        - System could be rapidly installed and commissioned throughout the world using local commissioning engineers.
        - System could be diagnosed from anywhere in the world via X-Panel keeping any potential disruption to an absolute minimum.

    2. Room Setup:

        - System could be configured via the touchpanel to match the specific system hardware, layout and furniture within the room.
        - System had x1 program and x1 touchpanel file for all deployments drastically simplifying deployment and overall management.

3. Codec:

- System would communicate with its local Cisco codec via a secure SSH connection as defined by the hostname entered into the touchpanel allowing the codec to be fully decoupled from the system.
- System had comprehensive codec interrogation and diagnostics capabilities available on touchpanel via the codec API.

4. Linear Development:

- Development was kept linear with no forking of the source code when additional functionality was requested by the client.
- Consequently it was not necessary to maintain a large record of versioning running on all systems throughout the world, if a processor or touchpanel needed replacing then just upload the latest version of the compiled file and the previous state of the system could be quickly restored.

- An initial x8 systems were deployed on the executive floor of the HSBC HQ, such that the feedback was so positive that they were enquiring on the price of x2000 systems within a few weeks.

- The systems were accepted by HSBC as the global standard eventually being deployed in 22 countries enabling AVMI to expand into the US and Asia.

Barclays Bank, Digital Signage Management and Control
UK Rollout

- Barclays branches throughout the UK contain a varying number of LCD's used for displaying digital signage.

- The bank required the ability to manage, schedule and control all LCD's within each branch via the banks LAN from a central location.

- The following was deployed in each branch:

- Crestron processor installed on the banks LAN running an X-Panel only.
- Standardised program and X-Panel layout deployed in all branches.
- Crestron program was configured via a *.csv file containing all system information e.g. LCD hostnames, schedules, groupings.
- Crestron program would automatically run a daily download of its specific *.csv file from an FTP server on the banks network.
- All *.csv files were centrally managed and automatically pulled down to the processor in each branch overnight.

Deutsche Bank, New York, x8 Boardrooms
Remote Commission

- AVMI had obtained the install and commission of x8 substantial boardrooms for Deutsche Bank in New York.

- Due to budgetary and logistical reasons it was necessary to commission all x8 AMX systems via remote, where a local engineer from AVMI's US office would attend site and I would upload and commission all x8 systems through the engineer's laptop.

- The systems included:

    - AMX 19.4" MXT panoramic touchpanels.
    - Client supplied touchpanel template containing substantial video conference, audio conference and presentation functionality.

- The overall process included:

    - Offsite Development: The touchpanel and source code were created.

    - Offsite Test: The touchpanel and source code were uploaded into the systems prior to install onsite.
                                Regular sessions were setup in order to maximise the time available for test given the time difference.
                                Good communication between the engineer and I enabled for smooth testing and progress.

    - Onsite Commission: The engineer would setup his laptop on the same network as the AMX systems, I would remote connect.
                                The systems were successfully installed and commissioned.

Touchpanel and Processor Source Code Traits

Below is a summary of some of the desired and undesired traits of all touchpanel UI and processor source code design:

- The traits are applicable to all areas of touchpanel design and source code implementation.
- The traits naturally combine, compound, promote, re-enforce, build-in and inherently work towards achieving successful and stable software.
- The traits should form the core considerations behind all touchpanel design and implementation decisions.

Touchpanel UI

Processor Source Code

Desired Traits:

Touchpanel UI		Processor Source Code	
Simple	Efficient	Simple	Symmetry
Organised	Smooth	Organised	Clean
Maintainable	Polished	Maintainable	Elegant
Extensible	Balanced	Extensible	Neat
Intuitive	Symmetry	Readable	(Appropriate) Restriction
Conventional	Closed Paths	Efficient	(Appropriate) Flexibility
Satisfaction	'Non-Technical'	Clarity (in design and structure)	Confidence
Familiar	'Home Button'	Clarity (in role)	Minimal Risk
Easy Mental Map		Naming Conventions	Modular
Ergonomic		Consistent	Cohesive
Transparent/Invisible Technology		Conscientious	Loose Coupling
		Discipline	Reassurance
		Safe	Motivation

Undesired Traits:

Touchpanel UI	Processor Source Code	
Frustration	Unnecessary layers of complexity	Verbosity/Ambiguity
Unresponsiveness	Unnecessary layers of processing/handling	Misleading comments/code
Impedance	Unnecessary generalisation	Redundant comments/code
Hinderance	Unnecessary feature anticipation	Rigidity
Clunkiness	Premature optimisation	Brittleness
Awkwardness	Over engineering	Clunky