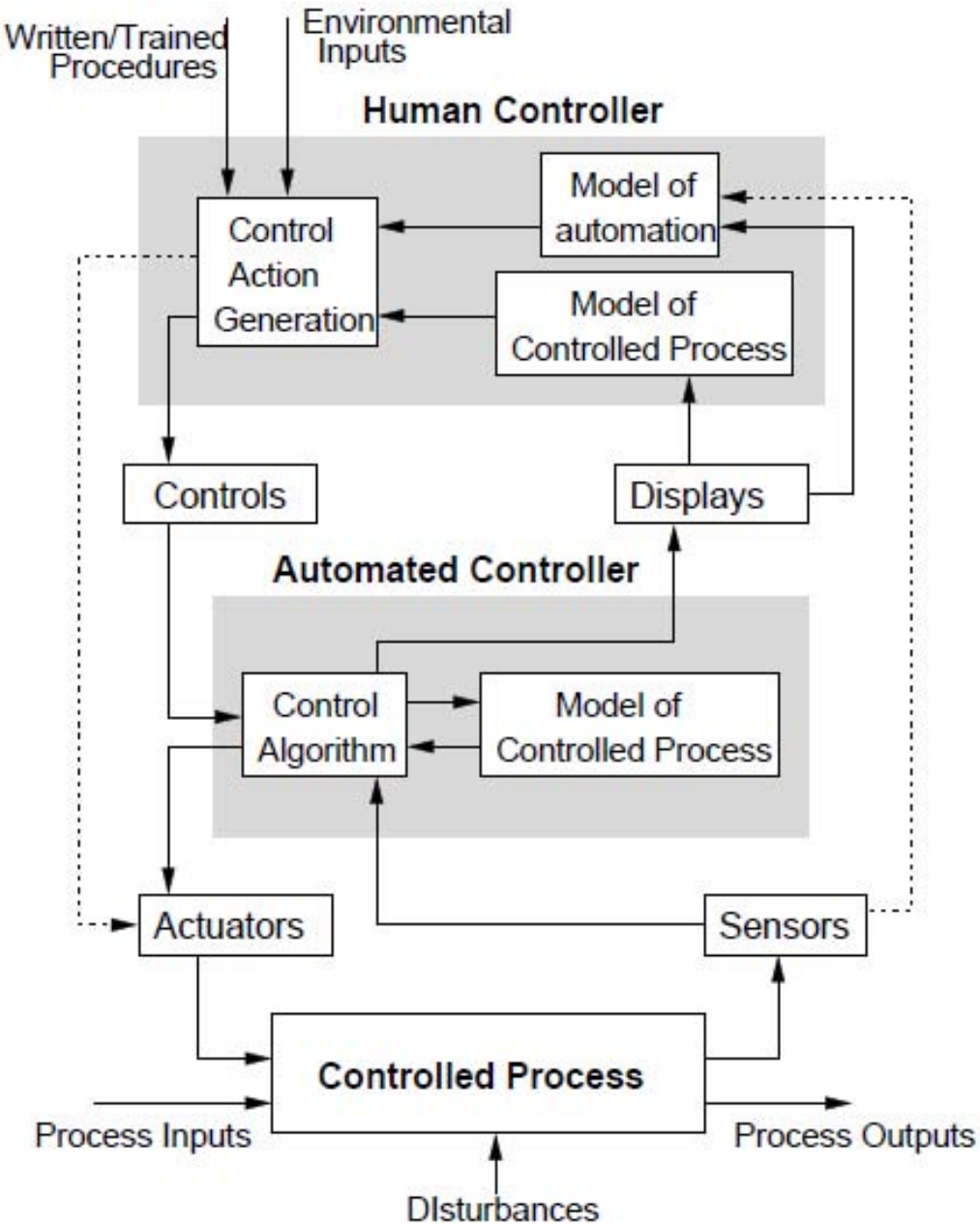
Safety-Guided Design

Process

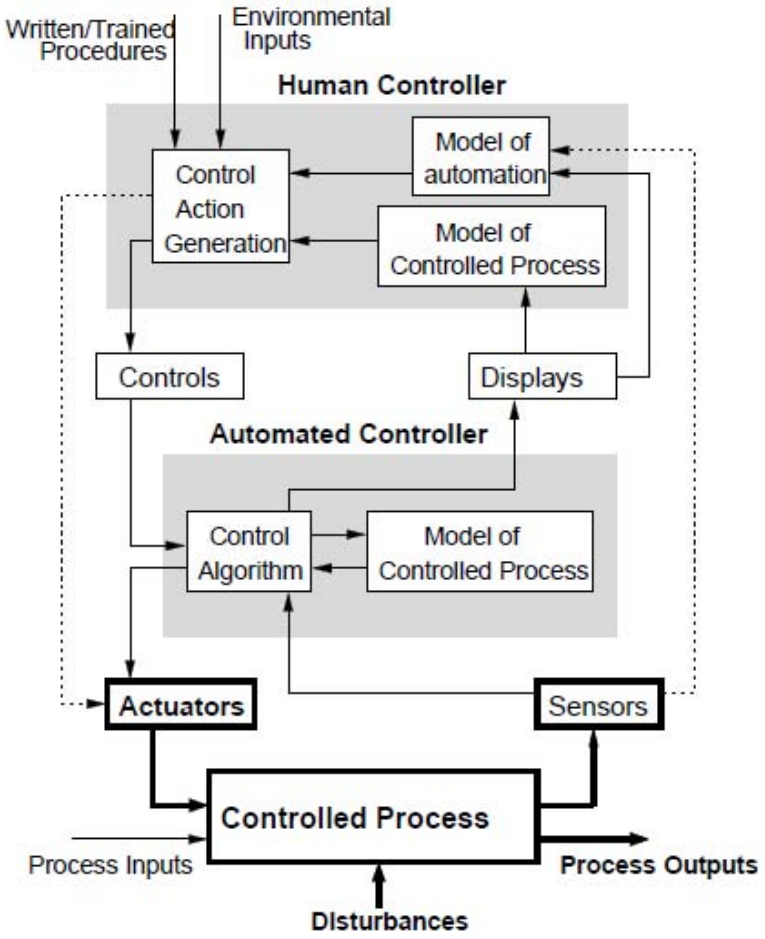
1. Try to eliminate hazards from conceptual design
2. If cannot eliminate, identify controls at system level
3. Create system control structure
4. Refine constraints and design in parallel
    a. STPA step 1: identify potentially hazardous control actions. Restate as design constraints.
    b. STPA step 2: determine factors that could lead to violation of safety constraints
    c. Augment basic design to eliminate or control
    d. Iterate and refine design

General Design for Safety Principles

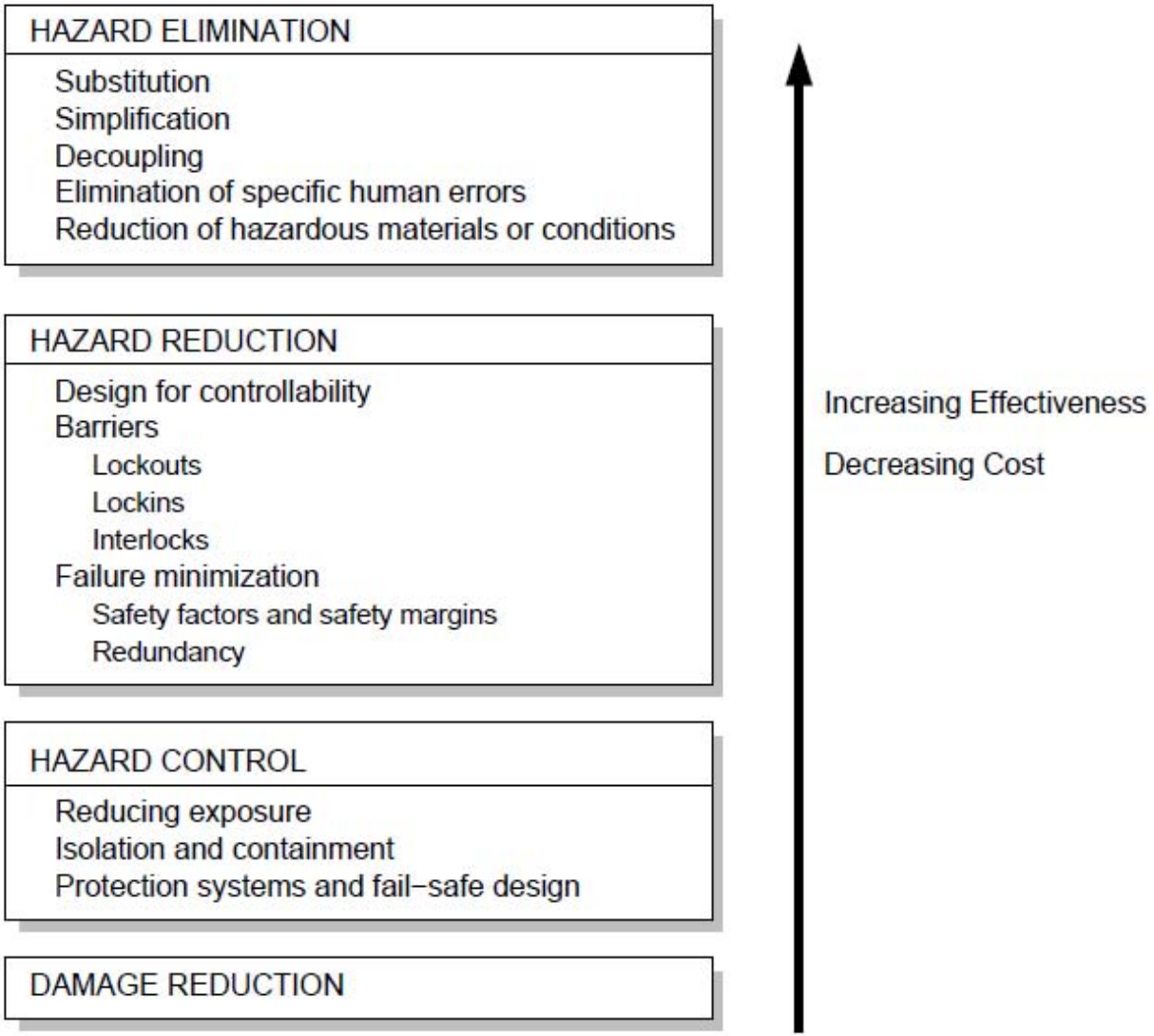
- In addition to identified application-specific design constraints
- Result from:
  - General STAMP principles of accident causation
  - General engineering design principles
  - Causes of past accidents
  - (requirements completeness criteria in *Safeware*)
- Divided into
  - General principles for any controller
  - Special system design principles for human controllers


Written/Trained Procedures
Environmental Inputs
Human Controller
Model of automation
Control Action Generation
Model of Controlled Process
Controls
Displays
Automated Controller
Control Algorithm
Model of Controlled Process
Actuators
Sensors
Controlled Process
Process Inputs
Process Outputs
DIsturbances

Controlled Process/Physical Component



Basic System Safety Design Precedence



Turbine Generator Example

Safety requirements:

1. Must always be able to close steam valves within a few hundred milliseconds
2. Under no circumstances can steam valves open spuriously whatever the nature of internal or external fault.

Divided into two parts (decoupled) on separate processors:

1. Non-critical functions: loss cannot endanger turbine nor cause it to shutdown.
   - Less important control functions
   - Supervisory, coordination, and management functions
2. Small number of critical functions

Turbine Generator Example (2)

- Uses polling: No interrupts except for fatal store fault (unmaskable)
  - Timing and sequencing thus defined
  - More rigorous and exhaustive testing possible
- All messages unidirectional
  - No recovery or contention protocols required
  - Higher level of predictability
- Self-checks of
  - Sensibility of incoming signals
  - Whether processor functioning correctly
- Failure of self-check leads to reversion to safe state through fail-safe hardware
- State table defines
  - Scheduling of tasks
  - Self-check criteria appropriate under particular conditions

Design for Controllability

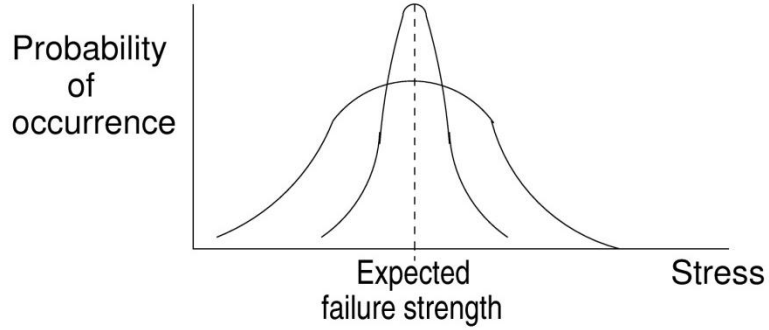
- Make system easier to control, both for humans and computers
  - Use incremental control
    - Perform critical steps incrementally rather than in one step
    - Provide feedback
      - To test validity of assumptions and models upon which decisions are made
      - To allow taking corrective action before significant damage is done
    - Provide various types of fallback or intermediate states
  - Lower time pressures
  - Provide decision aids

Monitoring

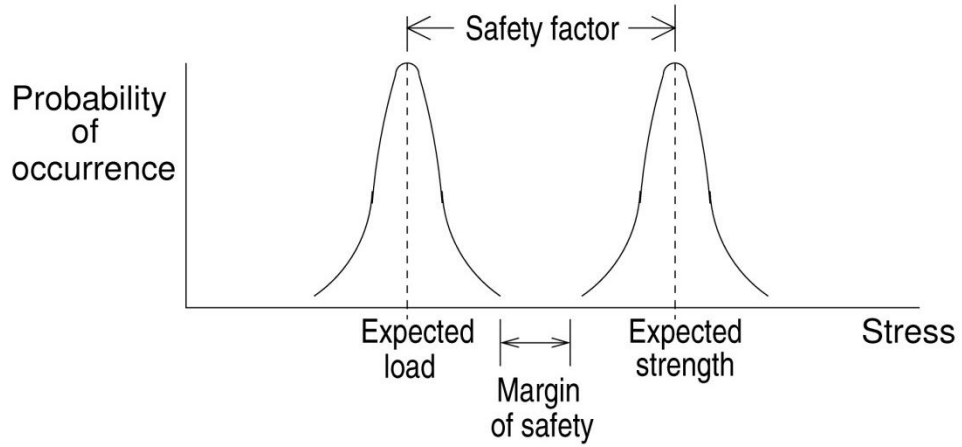
- Difficult to make monitors independent
  - Checks usually require access to information being monitored, but usually involves possibility of corrupting that information
  - Depends on assumptions about behavior of system and about errors that may or may not occur
    - May be incorrect under certain conditions
    - Common incorrect assumptions may be reflected both in design of monitor and devices being monitored.

Failure Minimization:
Safety Factors and Safety Margins

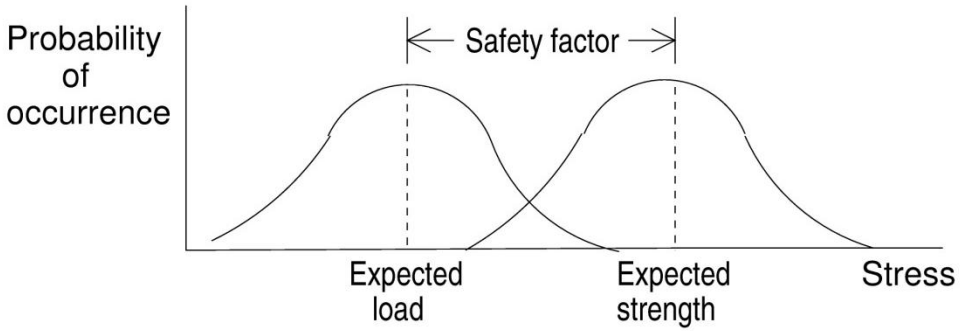
- Used to cope with uncertainties in engineering
  - Inaccurate calculations or models
  - Limitations in knowledge
  - Variation in strength of a specific material due to differences in composition, manufacturing, assembly, handling, environment, or usage.
- Some ways to minimize problem, but cannot eliminate it
- Appropriate for continuous and non-action systems



(a) Probability density function of failure for two parts with same expected failure strength.



(b) A relatively safe case.



(c) A dangerous overlap but the safety factor is the same as in (b)

Failure Minimization: Redundancy

- Goal is to increase reliability and reduce failures
  - Common-cause and common-mode failures
  - May add so much complexity that causes failures
  - More likely to operate spuriously
  - May lead to false confidence (Challenger)
- Useful to reduce hardware failures. But what about software?
  - Design redundancy vs. design diversity
  - Bottom line: Claims that multiple version software will achieve ultra-high reliability levels are not supported by empirical data or theoretical models

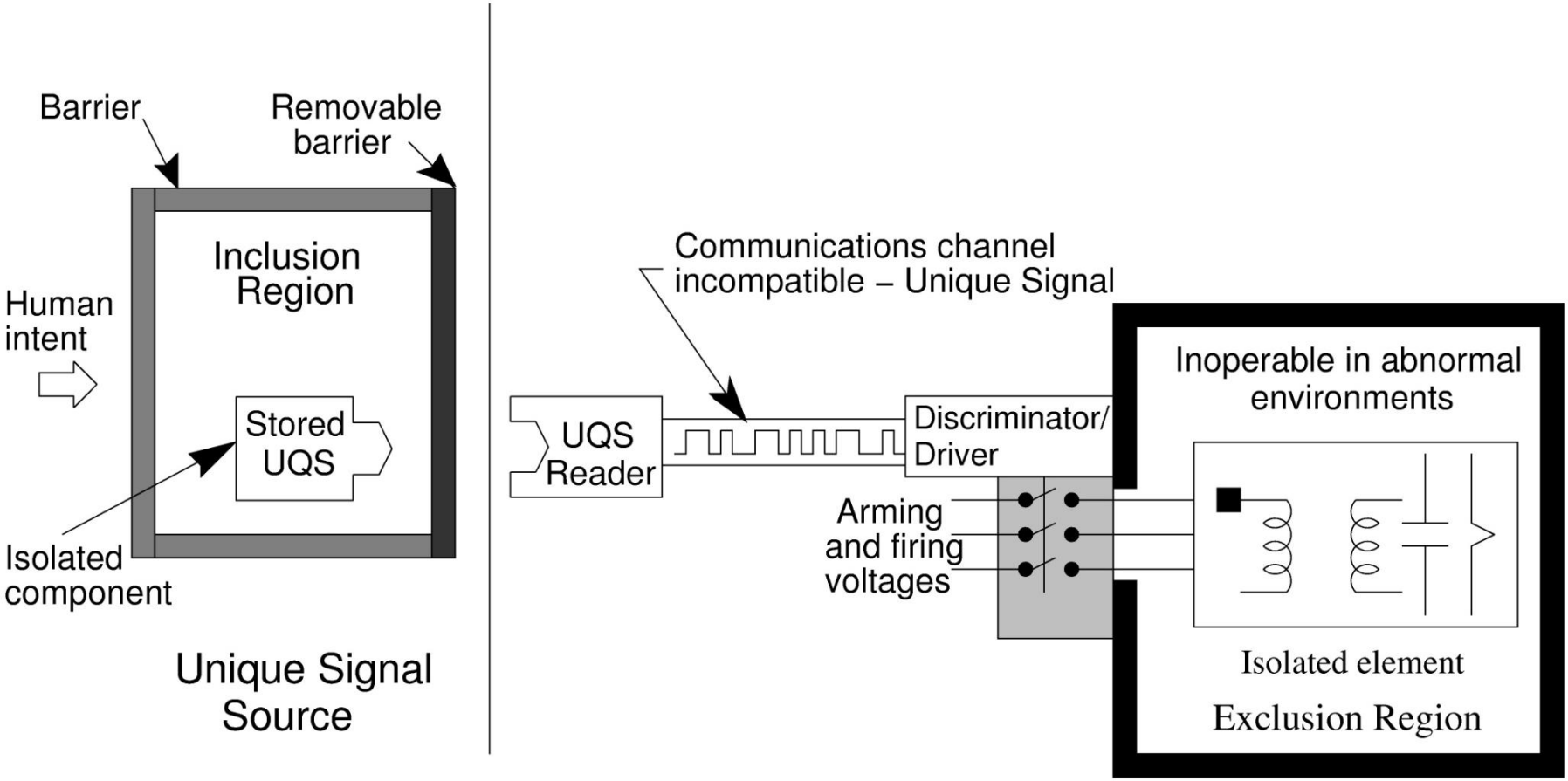
Example: Nuclear Detonation

- Safety depends on NOT working
- Three basic techniques (callled “positive measures”)
  1. Isolation
     - Separate critical elements
  2. Inoperability
     - Keep in inoperable state, e.g., remove ignition device or arming pin
  3. Incompatibility
     - Detonation requires an unambiguous indication of human intent be communicated to weapon
     - Protecting entire communication system against all credible abnormal environments (including sabotage) not practical.
     - Instead, use unique signal of sufficient information complexity that unlikely to be generated by an abnormal environment

Example: Nuclear Detonation (2)

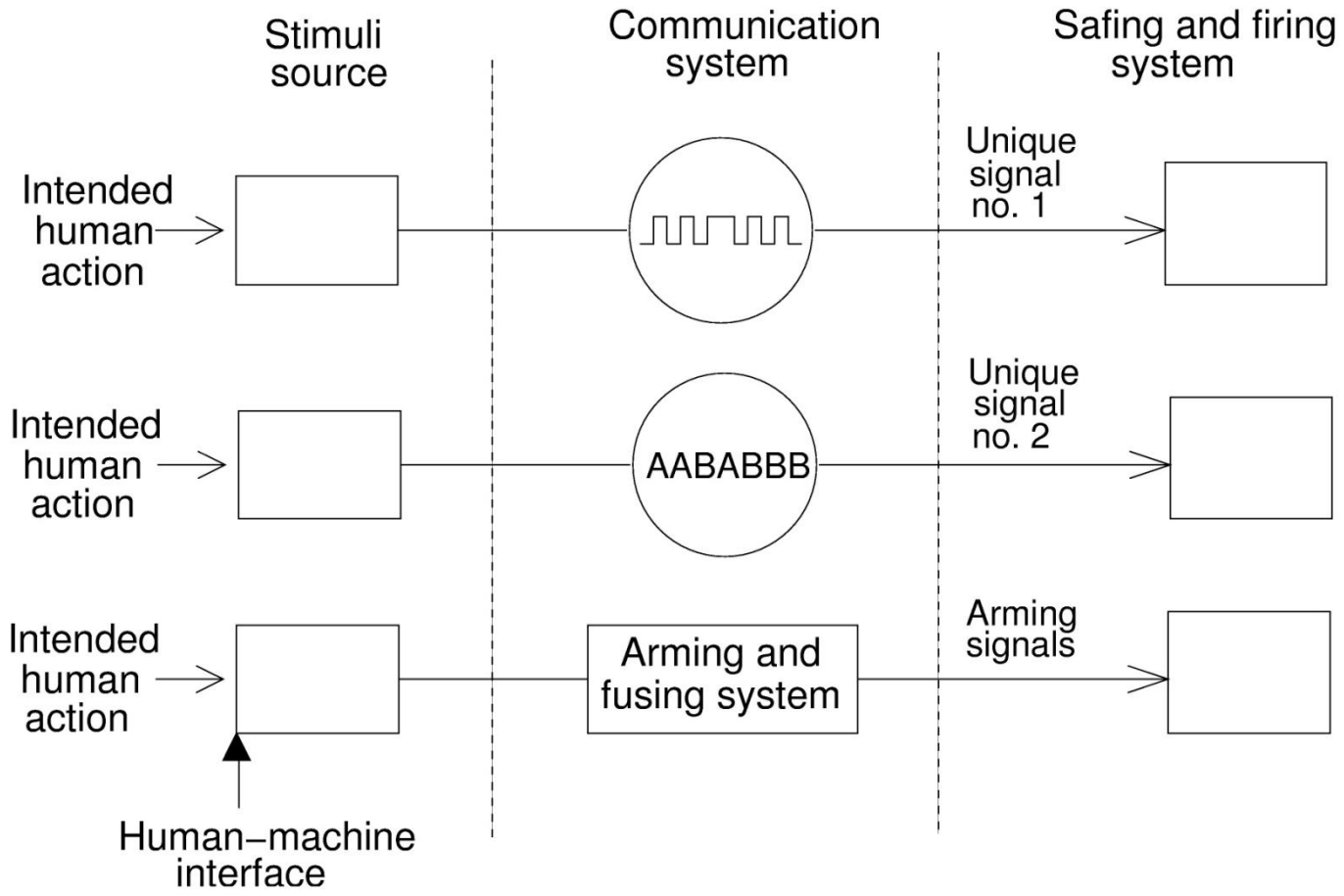
- Unique signal discriminators must
  1. Accept proper unique signal while rejecting spurious inputs
  2. Have rejection logic that is highly immune to abnormal environments
  3. Provide predictable safe response to abnormal environment
  4. Be analyzable and testable
- Protect unique signal sources by barriers
- Removable barrier between these sources and communication channels

Example: Nuclear Detonation (3)



Example: Nuclear Detonation (4)

May require multiple unique signals from different individuals along various communication channels, using different types of signals (energy and information) to ensure proper intent.



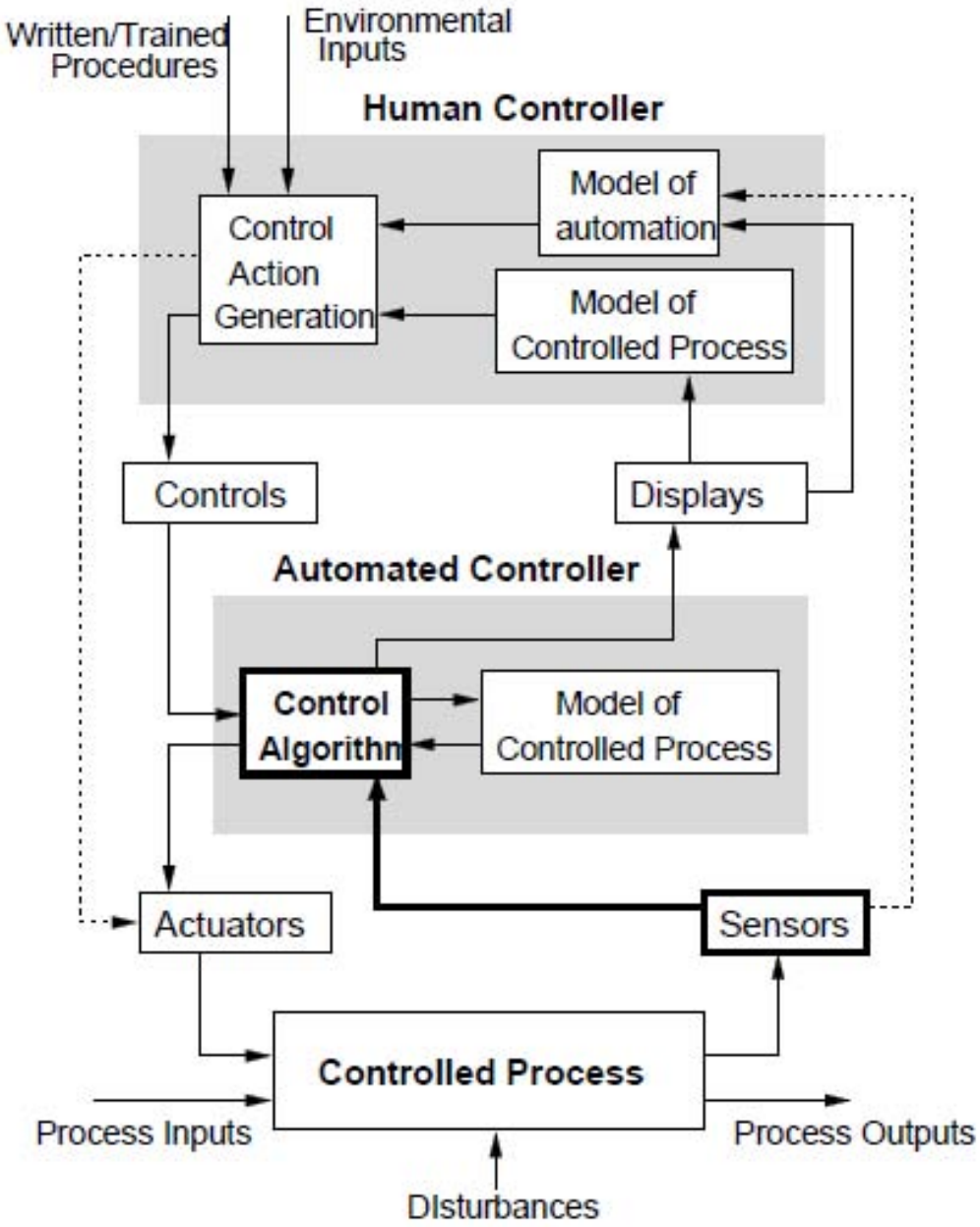
Protection Systems and Fail-Safe Design

- Depends upon existence of a safe state and availability of adequate warning time
- May have multiple safe states, depending on process conditions
- General rule is hazardous states should be hard to get into and safe states should be easy
- Panic button

Protection Systems and Fail-Safe Design (2)

- Watchdog timer: Software it is protecting should not be responsible for setting it
- Sanity checks (I'm alive signals)
- Protection system should provide information about its control actions and status to operators or bystanders.
- The easier and faster is return of system to operational state, the less likely protection system is to be purposely bypassed or turned off

Designing and Processing Inputs and Feedback



Designing and Processing Inputs and Feedback

- STPA provides information about what types of feedback needed
- Additional general design principles:
  - Design to respond appropriately to arrival of any possible input at any time and lack of expected input over a given time period.

    (e.g., target detection report from shutdown radar)
  - Check all inputs for out-of-range or unexpected values. Design response into control algorithm.
  - Specify max time computer waits until before first input and what to do if violated

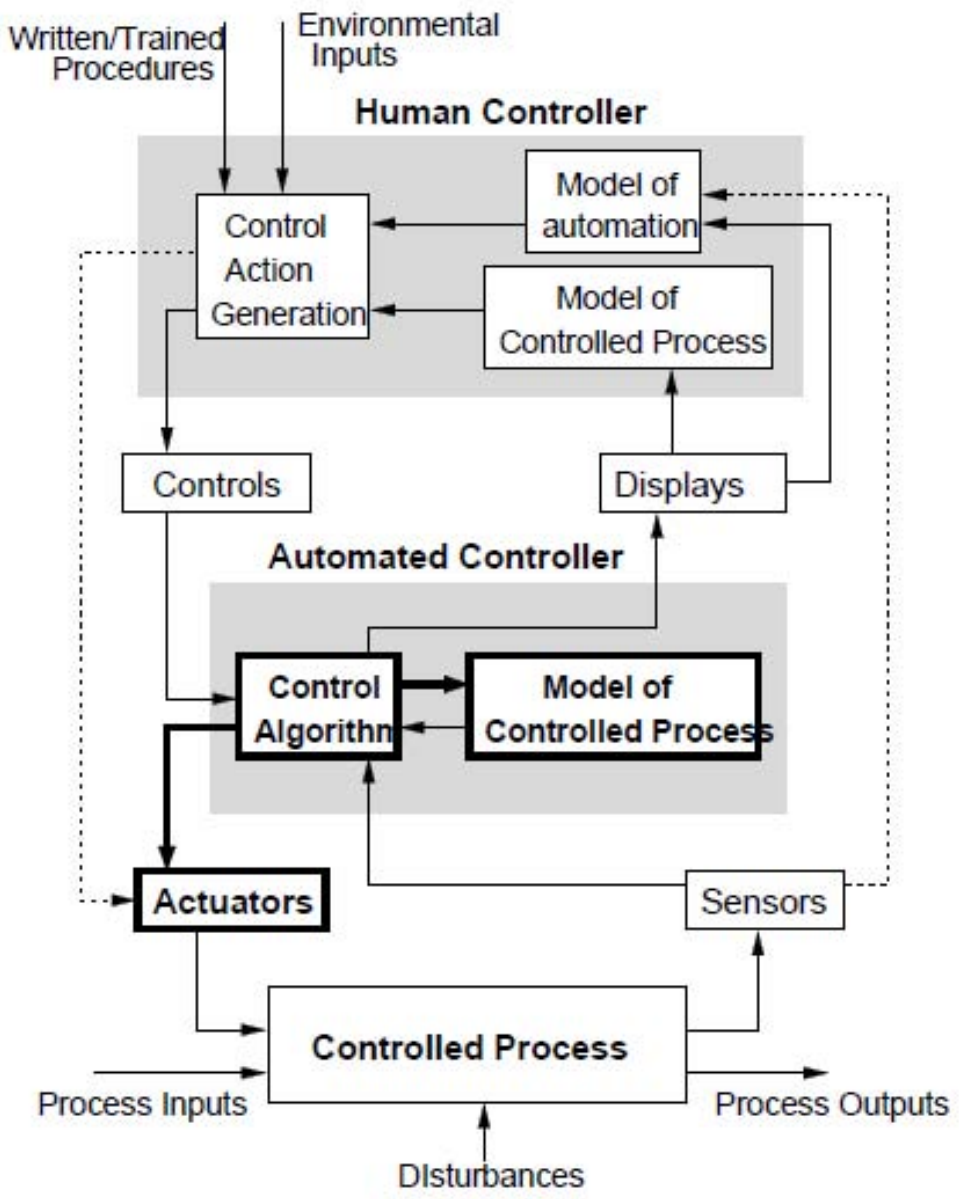
Designing and Processing Inputs and Feedback (2)

- Time bounds (min and max) should be checked for every input and appropriate behavior provided in case does not arrive within bounds.
- Specify response for non-arrival of an input (timeout) and excessive inputs (overload condition)
- Minimum arrival check for each physically distinct communication path (sanity or health check). Software should have the capability to query its environment with respect to inactivity over a given communication path

Feedback Loops

- Basic feedback loops, as defined by the process control function, must be included in algorithm along with appropriate checks to detect internal or external failures or errors.
- There should be an input that the software can use to detect the effect of any output on the process.
  - Not just that command arrived but actual execution
- Every output to which a detectable input is expected must have associated with it:
  1. Behavior to handle the normal response
  2. Behavior to handle a response that is missing, too late, too early, or has an unexpected value.

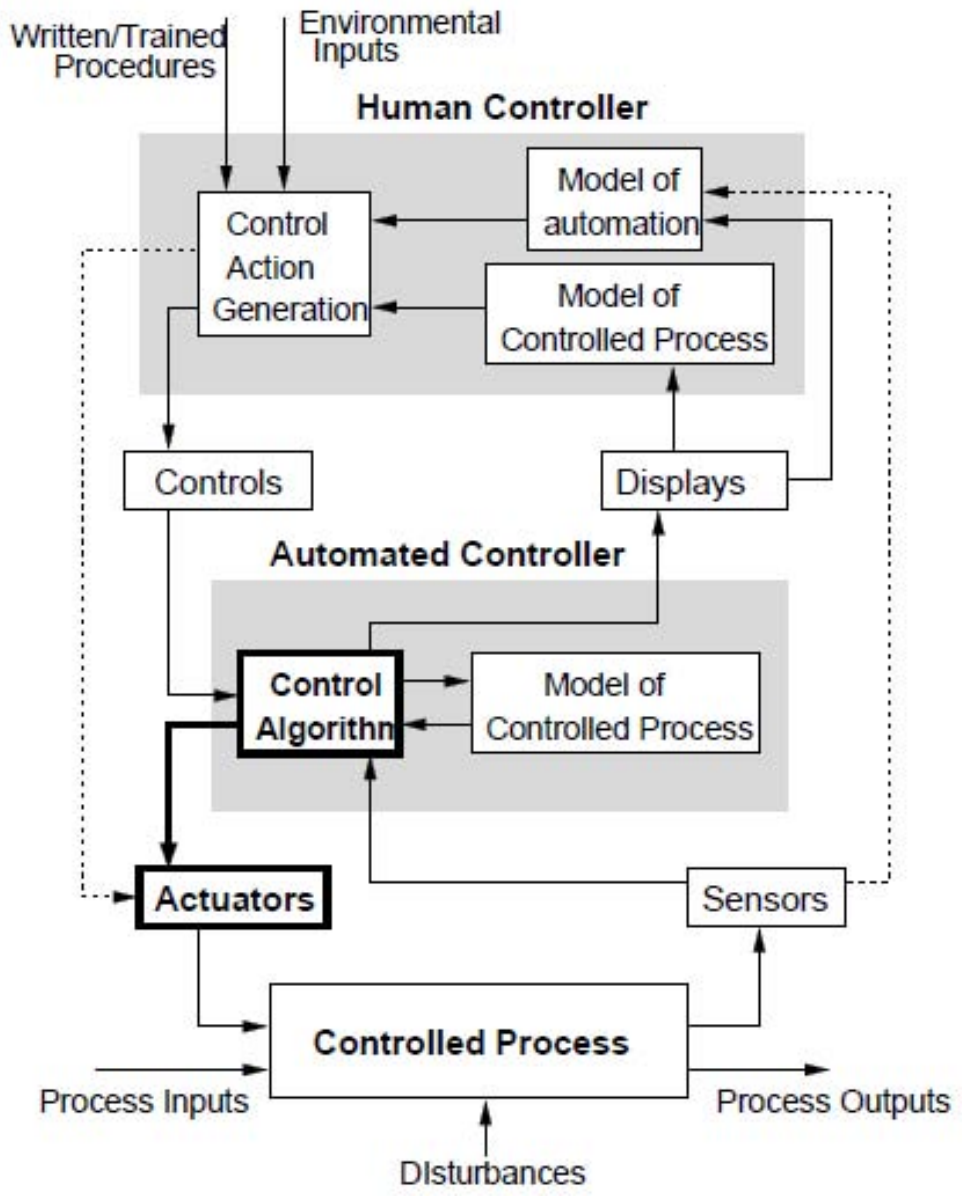
Initializing and Updating the Process Model



Initializing and Updating Process Model

- Process model must reflect actual process state at initial startup and after temporary shutdown.
  - Unknown state
- Must start in a safe state. Interlocks should be initialized or checked to be operational at system startup, including startup after temporarily overriding interlocks.
- Behavior of software with respect to inputs received before startup, after shutdown, or when computer is temporarily disconnected from process (off-line) must be specified or it must be determined that this information can be safely ignored.
  - National Flight Data Center (led to airport charts inconsistent with reality)

Producing Control Outputs



Producing Outputs

- For the largest interval in which both input and output loads are assumed and specified, the absorption rate of the output environment must equal or exceed the input arrival rate.
- Contingency action must be specified when the output absorption rate limit is exceeded.
- Behavior should be deterministic: only one behavior specified for arrival of any input in a particular state.

Data Age

- All inputs used in specifying output events must be properly limited in the time they can be used
- Output commands that may not be able to be executed immediately must be limited in the time they are valid.
- Incomplete hazardous action sequences (transactions) should have a finite time specified after which the software should be required to cancel the sequence automatically and inform the operator.
- Revocation of partially completed transactions may require:
  1. Specification of multiple times and conditions under which varying automatic cancellation or postponement actions are taken without operator confirmation
  2. Specification of operator warnings to be issued in case of such revocation

Latency Criteria

- Latency is the time interval during which receipt of new information cannot change an output even those it arrives prior to the output
  - Influenced by hardware and software design (e.g., interrupt vs. polling)
  - Cannot be eliminated completely
  - Acceptable length determined by controlled process

Fault Handling

Need to handle:

- Off-nominal states and transitions
- Performance degradation
- Communication with operator about fail-safe behavior
- Partial shutdown and restart (paths to and from fail-safe states)
- Hysteresis in transitions between off-nominal and nominal
  - Conditions that caused it to leave normal state may still exist
- Failure into safe state

Hazard-Reducing vs. Hazard-Increasing Outputs

- **Soft failure mode**: loss of ability to receive input X *could* inhibit production of output command Y
- **Hard failure mode**: loss of ability to receive input X *will* inhibit production of output Y
- Soft and hard failure modes should be eliminated for all hazard-reducing outputs
  - Multiple ways should be provided for triggering commands that maintain safety.
- Hazard-increasing output should have both soft and hard failure modes.
  - Multiple inputs or triggers should be required for triggering commands that can lead to hazardous states.

Designing for Human Control

Advantages of Humans

- Human operators are adaptable and flexible
  - Able to adapt both goals and means to achieve them
  - Able to use problem solving and creativity to cope with unusual and unforeseen situations
  - Can exercise judgment
- Humans are unsurpassed in
  - Recognizing patterns
  - Making associative leaps
  - Operating in ill-structured, ambiguous situations
- Human error is the inevitable side effect of this flexibility and adaptability

Role of Humans in Automated Systems

- The Human as Monitor
  - Task may be impossible
  - Dependent on information provided
  - Difficult (impossible?) to monitor for infrequent events
  - State of information more indirect
  - Failures may be silent or masked
  - Little active behavior can lead to lower alertness and vigilance, complacency, and over-reliance

Role of Humans in Automated Systems (2)

- The Human as Backup
  - May lead to lowered proficiency and increased reluctance to intervene
  - Limited ability to practice to handle “breakdown” scenarios
  - Fault intolerance may lead to even larger errors
  - May make crisis handling more difficult

Role of Humans in Automated Systems (3)

- The Human as Partner
  - May be left with miscellaneous tasks
  - Tasks may be more complex and new tasks added
  - By taking away easy parts, may make difficult parts harder
  - Problems in communication between humans and automation

Consequences of Computers

- High tech automation changing cognitive demands on operators
  - Supervising rather than directly monitoring
  - More cognitively complex decision-making
  - Complicated, mode-rich systems
  - Increased need for cooperation and communication
- Human-factors experts complaining about technology-centered automation
  - Designers focus on technical issues, not on supporting operator tasks
  - Leads to “clumsy” automation

Mixing Humans and Computers

- Automated systems on aircraft have eliminated some types of human error and created new ones
  - Errors of commission vs. errors of omission
- Human skill levels and required knowledge may go up
- Correct partnership and allocation of tasks is difficult

  Who has the final authority?
- Authority limits
  - Prevent actions that would lead to hazardous states but
  - May prohibit maneuvers needed in extreme situations.

Incidents related to Operator Authority Limits

- *Warsaw*
- *LAX incident*:
  - During one A320 approach, pilots disconnected the autopilot while leaving the flight director engaged.
  - Under these conditions, the automation provides automatic speed protection by preventing aircraft from exceeding upper and lower airspeed limits.
  - At some point during approach, after flaps 20 had been selected, the aircraft exceeded the airspeed limit for that configuration by 2 kts. As a result, the automation intervened by pitching the aircraft up to reduce airspeed back to 195 kts.
  - The pilots, who were unaware that automatic speed protection was active, observed the uncommanded automation behavior. Concerned about the unexpected reduction in airspeed at this critical phase of flight, they rapidly increased thrust to counterbalance the automation. As a consequence of this sudden burst of power, the aircraft pitched up to about 50 degrees, entered a sharp left bank, and went into a dive.
  - The pilots eventually disengaged the autothrust system and its associated protection function and regained control of the aircraft.

Typical Problems with IT

- Getting lost in display architecture
  - Difficult to find right page or data set
- Not coordinating computer entries among multiple people entering things
- Workload
  - Often increase demand at time when already a lot to do
  - Heads down work in aircraft
- Data overload, “keyhole problem”
  - May have to sort through large amounts of data to find pieces that reveal true nature of situation
  - Then need to integrate information
- Digital displays may require extra mental processing
  - Hard to notice changes (events, trends) with digital values clicking up and down

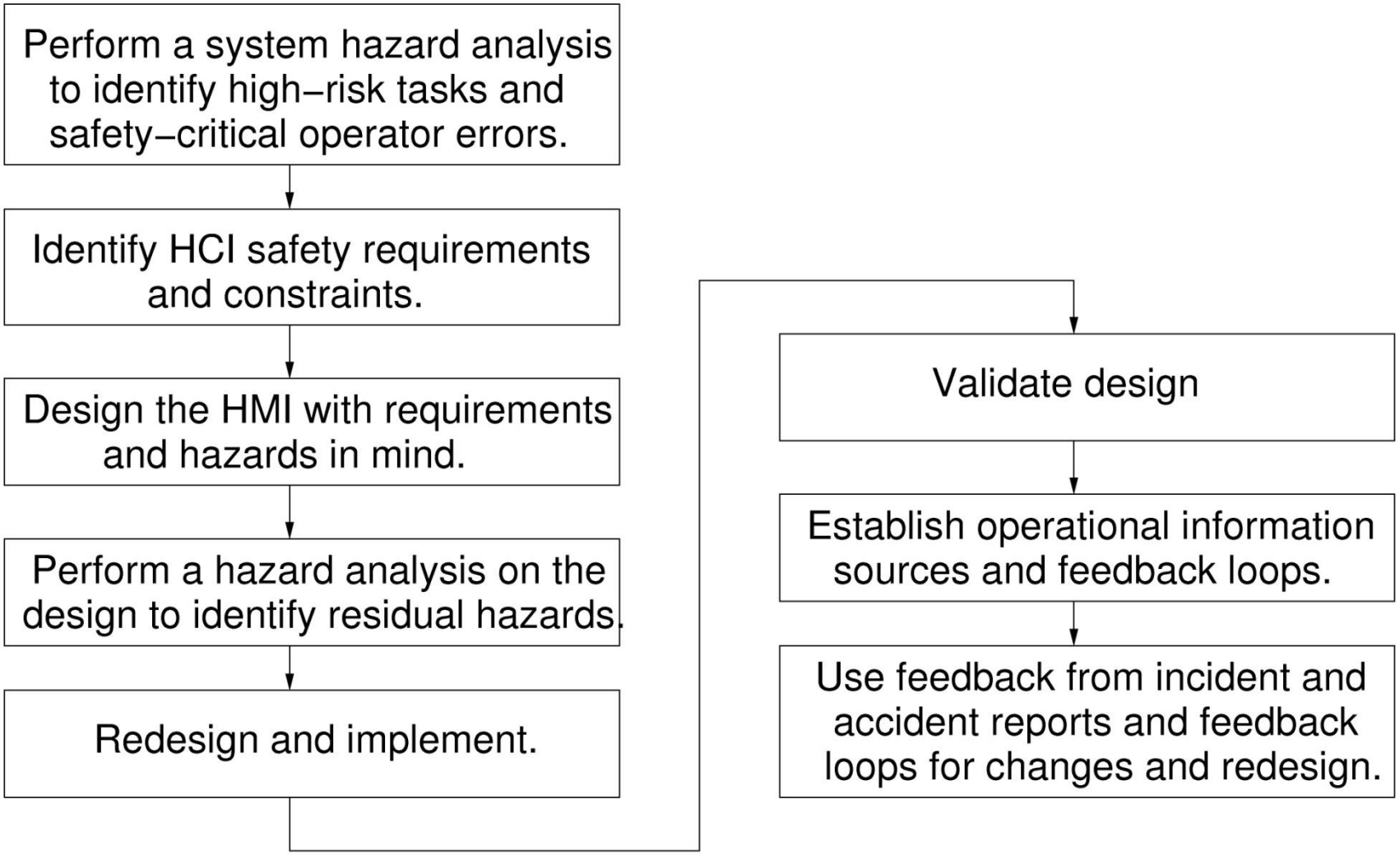
Cognitive Consequences of Computers

- Increase memory demands
- New skill and knowledge demands
- Can complicate situation assessment
- Can undermine people's attention management
- Can disrupt efficient and robust scanning patterns
- Can lead to limited visibility or changes and events, alarm and indication clutter, extra interface management tasks
- By increasing system reliability, can provide little opportunity to practice and maintain skills for managing system anomalies
- Force people into using tricks necessary to get task done that may not work in uncommon situations.

Designing for Human Control

- Human error is not random. It is systematically connected to features of peoples tools, tasks, and operating environment.
- Two ways to assist in human factors design:
  - Use hazard analysis (STPA) to provide information for human-machine interaction and interface design.
  - Apply general design principles based on what is known about human factors

HMI Design Process



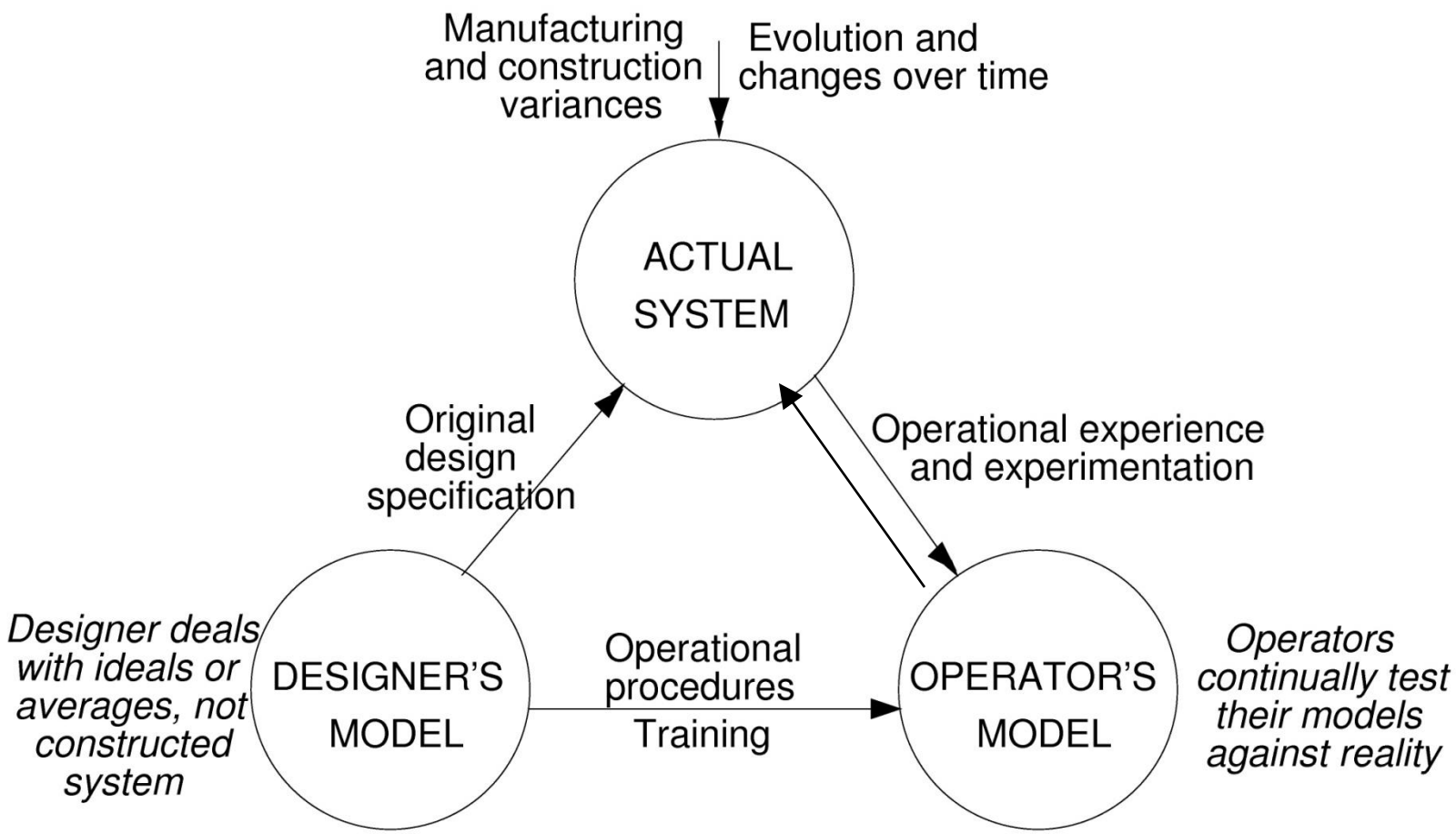
Establishing Operational Feedback

- Changes in practice become established norm over time. Lack of adverse consequences tends to affirm system is safe.
  - Need performance audits.
  - Starts from assumptions of hazard analysis and responsibilities assigned to controller

Human Error Fundamentals

- Slips vs. mistakes (Norman, Reason)
- Human error vs. human ingenuity
  - Rasmussen: Human error results from unsuccessful experiments in an unkind environment
  - Rasmussen: balance between optimizing skills and willingness to accept risk of exploratory acts
- Applying solutions that worked in other situations
- Designers can provide assistance to human problem solving (e.g., ways to test hypotheses safely)

Updating Process Models



Human Factors in Accidents

- Bounded rationality: people in operational work do not have unlimited time or resources
  - Have multiple inputs, multiple tasks, multiple goals
- When person focused on one set of features, less prominent features are unlikely to be detected (DeKeyser and Woods)
- People's understanding of a situation develops hand-in-hand with unfolding circumstances.

Human Factors in Accidents (2)

Cognitive fixation vs. Thematic Vagabonding

- Have to come up with a plausible explanation from emerging mass of uncertain, incomplete, and contradictory evidence.
  - Preliminary explanation allows setting on a plausible explanation for the data observed.
  - But can create preliminary hypotheses and trouble-shooting activities at expense of others.
- Cognitive Fixation: hold on to an assessment of a situation while new evidence about situation comes in
- Thematic vagabonding: Jumping from explanation to explanation, driven by loudest or latest indication or alarm.
- Only hindsight can show whether should have abandoned one explanation in favor of another or should have settled on a stable interpretation instead of just pursuing latest clue (Dekker)

Human Factors in Accidents (3)

- Plan Continuation (Orisanu): Sticking with an original plan while situation has actually changed and calls for a different plan.
  - Early cues suggesting original plan are usually very strong and unambiguous. Helps lock people into plan.
  - Later cues suggesting should be abandoned are typically fewer, more ambiguous, and not as strong. Conditions may deteriorate gradually. Evidence may not be compelling without hindsight.
  - Abandoning plan may be costly (entail organizational and economic consequences)
  - Challenge is understanding why it made sense to continue with their original plan. Which cues did they rely on and why?

Stress

Caused by demand-resource mismatch

- Tunneling: tendency to see and increasingly narrow focus of one's operating environment
  - Comes from human strength to form a stable and robust idea of a shifting world with multiple threads that compete for attention and where evidence may be uncertain and incomplete
- Regression: tendency to revert to earlier learned routines even if not entirely appropriate to current situation.
  - Frees up mental resources, do not have to match current perception with consciously finding out what to do each time

Stress (2)

- Distortion of time perspective under high workload and stress.
  - Fewer mental resources available to keep track of time
- Fatigue: Impairs judgment about how fatigued they are and how it affects their performance.
- Stress and fatigue encourage plan contination
  - More difficult to entertain multiple hypotheses
  - More difficult to project a situation into future by mentally stimulating effects of various decision alternatives.

Human Factors in Accidents (4)

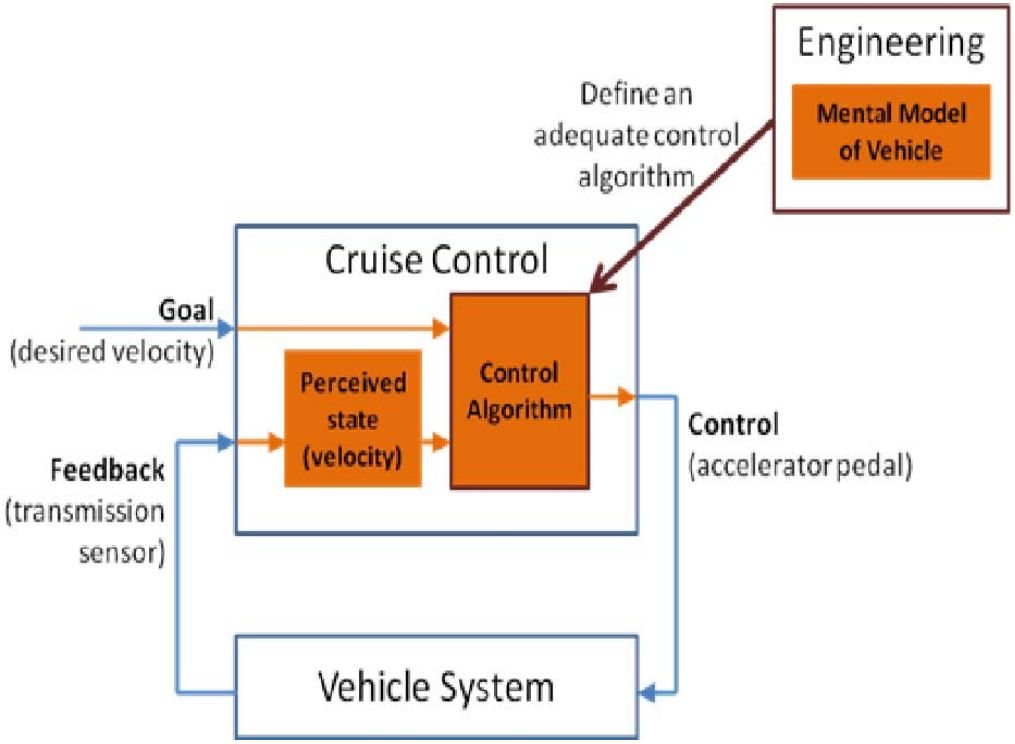
- Errors of omission vs. errors of commission (Sarter and Woods)
  - Related to changing role of humans in systems (supervisors of automated controllers)
  - Cognitive demands may not be reduced but simply change in their basic nature
  - Reduce some types of human errors but introduce new ones

General Design Principles

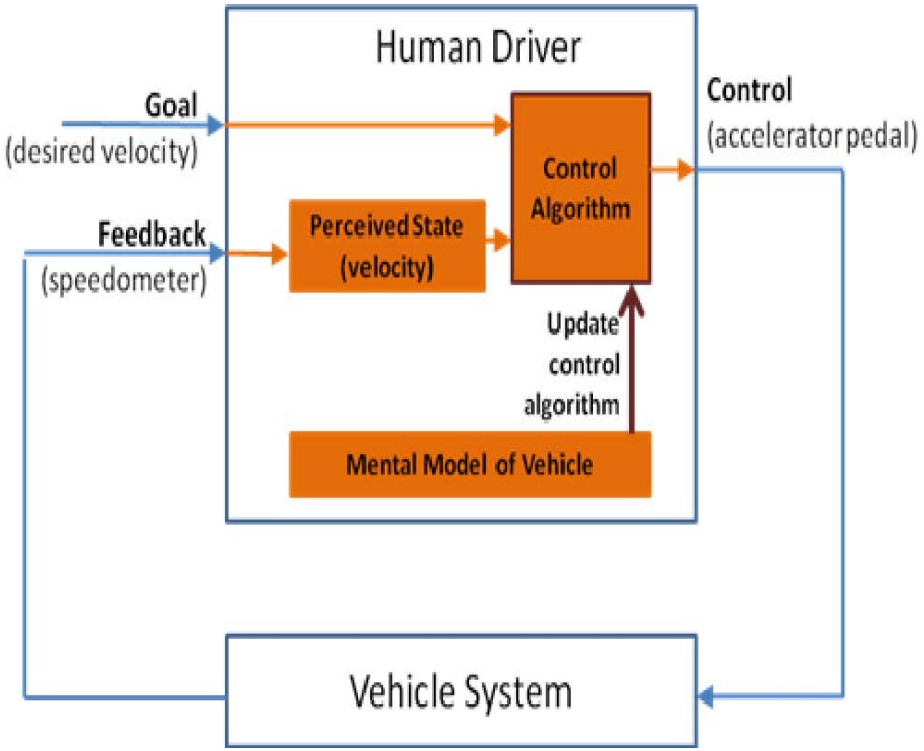
- If expect operators to react correctly to emergencies, need to design to support them and help fight tendencies described previously (e.g., cognitive fixation, tunneling, plan continuation)
- Design HMI to augment human abilities, not replace them (to aid the operator, not take over)
- Begin the design process by considering the operator and continue that perspective throughout
  - Involve operators in design decisions and safety analysis throughout development.

Physical vs. Human Controllers

Physical controllers have fixed control algorithms



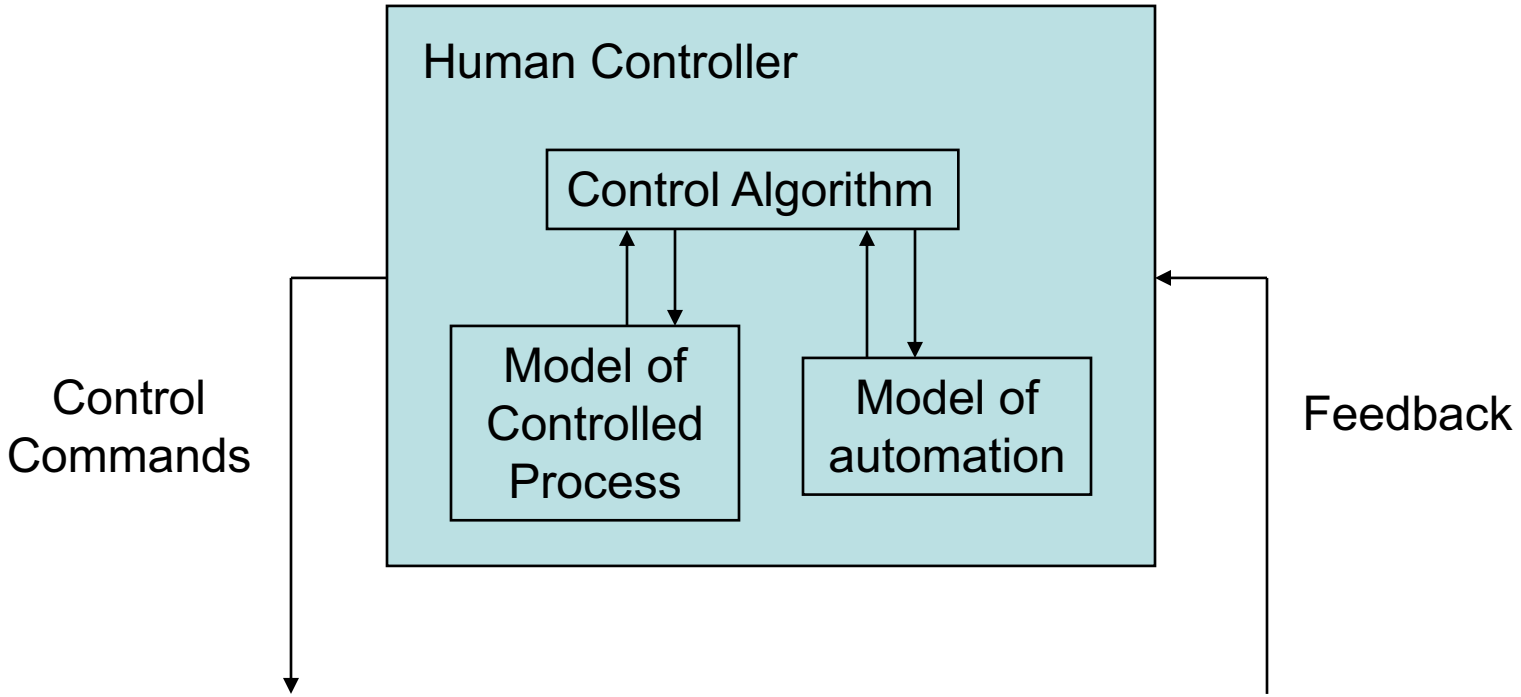
Human controllers have dynamic control algorithms (John Thomas)



Implications of Differences

- Most of STPA still holds.
- Analyze control algorithm as defined in procedures or control requirements
- New task of understanding and preventing unsafe changes in control algorithm.
  - Starts from basic human factors principles
  - Design process to lessen impact of common factors
  - Provide training on hazards and hazardous behavior, reasons for design, etc.
  - Audit practices to detect unsafe changes

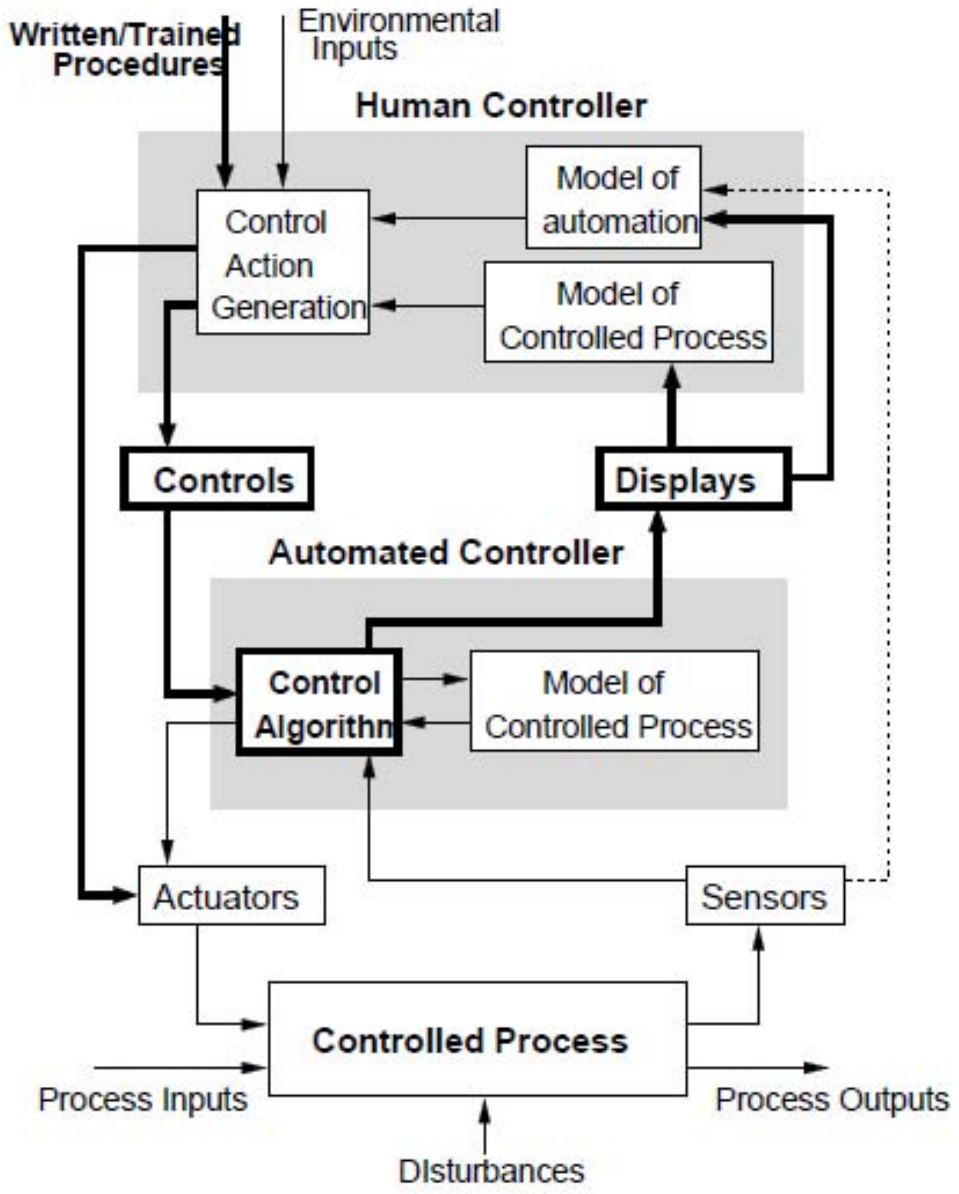
Human may also need model of automation



Has implications for training and for design of automation. Common problems:

- inconsistent behavior
- unintended side effects

Providing Control Options



Providing Control Options

- Avoid designs that require or encourage management by exception.
- Operators must have adequate flexibility to cope with undesired behavior and not be constrained by inadequate control options (Rasmussen)

Providing Control Options

- Design for incremental control:
  - Can observe controlled process and get feedback about previous steps.
  - Can modify or abort control actions before significant damage done
  - Must provide operator with compensating actions for incremental actions that have undesired effects.
- Provide multiple ways to change from an unsafe to a safe state.
- Provide multiple physical devices and logical paths to ensure that a single hardware failure or software error cannot prevent operator from taking action to maintain a safe system state and avoid hazards.

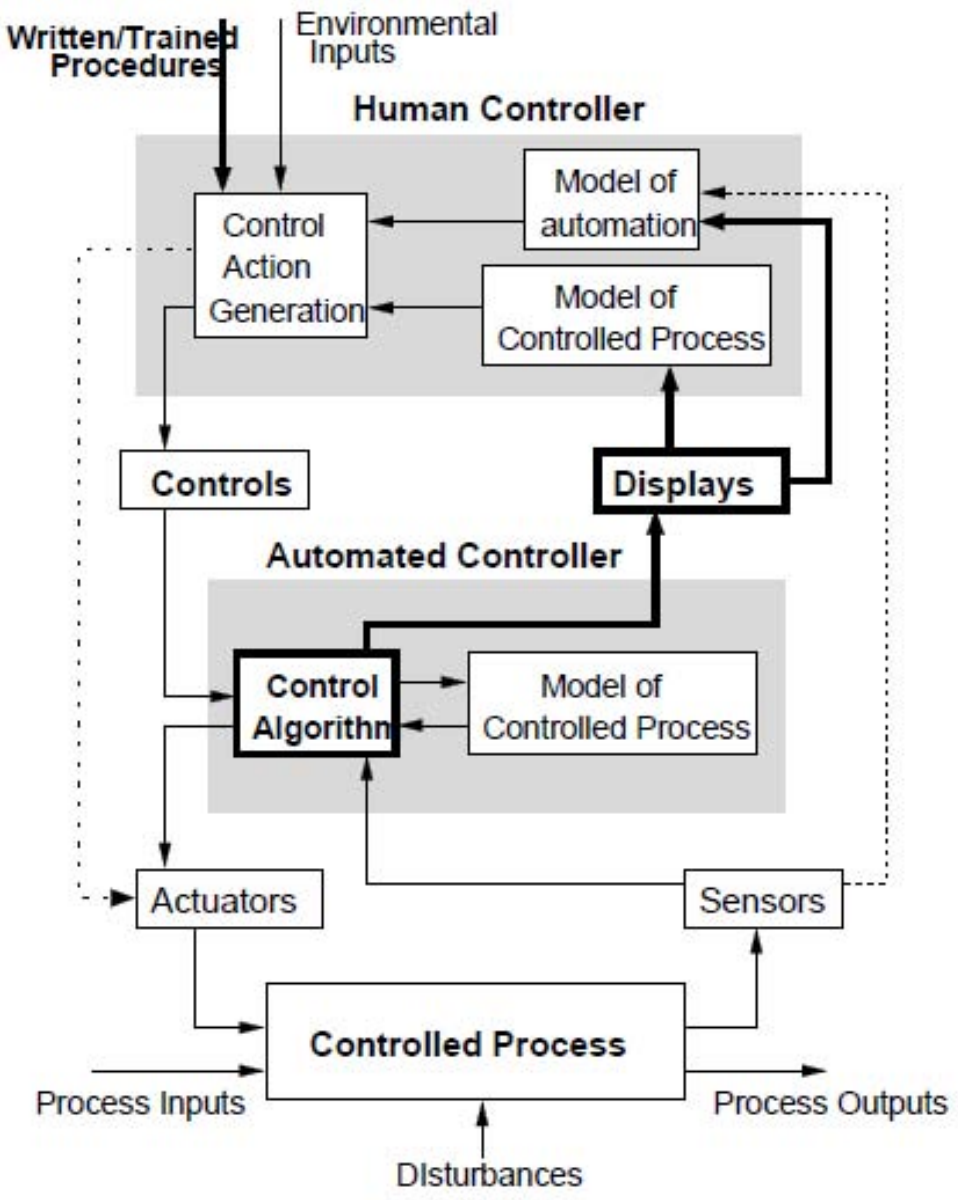
Design for Error Tolerance

- Many systems limit people's ability to detect and recover from their errors.
- In error tolerant systems:
  - Errors are observable (within an appropriate time limit)
  - Errors are reversible before unacceptable consequences
- Same true for computer errors: make them observable and reversible
- In general, allow controllers to monitor their own performance

Design for Error Tolerance (2)

- To design for error tolerance:
  - Help operators monitor themselves and recover from errors
  - Provide feedback about actions operators took (in case inadvertent, e.g., echoing back operator inputs and requiring confirmation) and effects
  - Allow for recovery from erroneous actions
    - Control options (compensating or reversing actions) and
    - Time for recovery actions to be taken

Matching Tasks to Human Characteristics



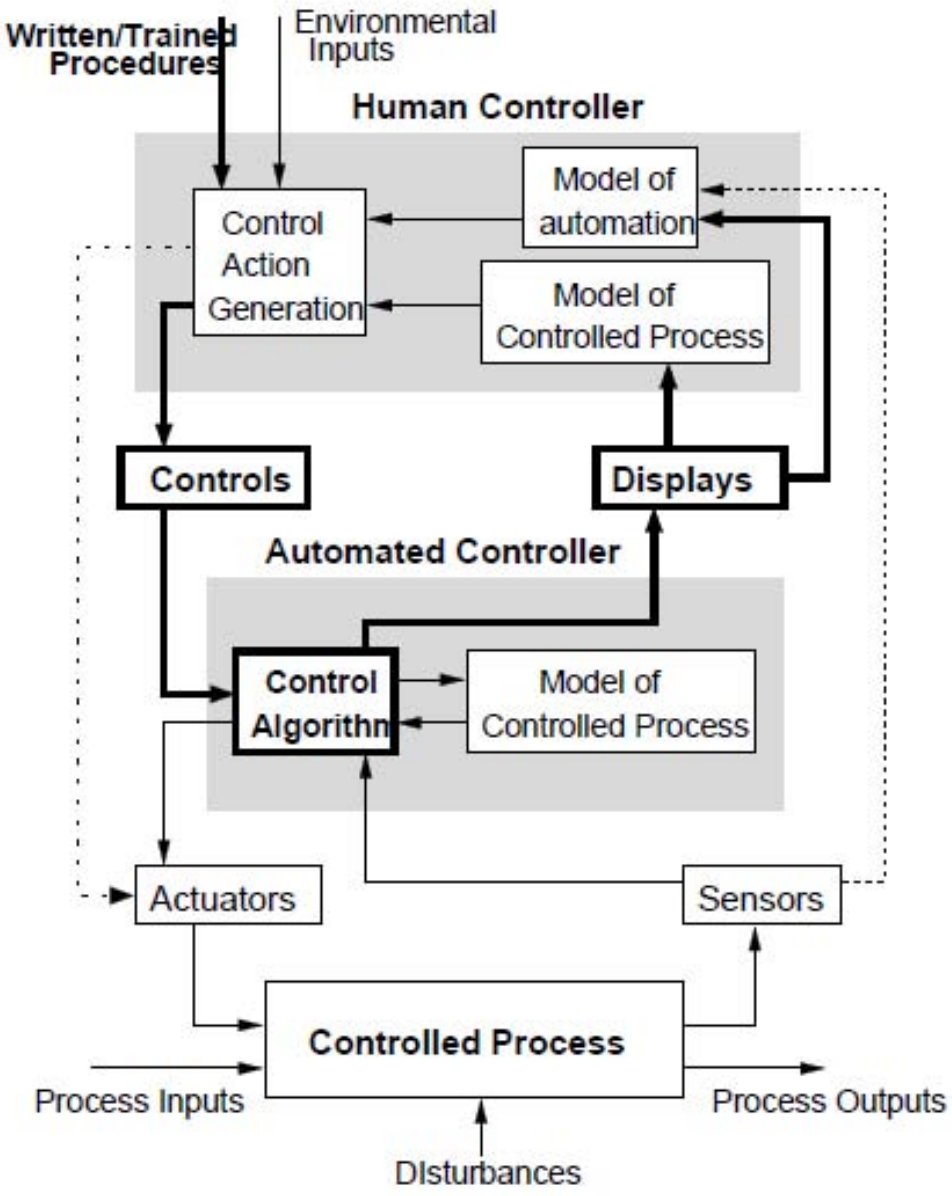
Matching Tasks to Human Characteristics

- Tailor systems to human requirements instead of vice versa
- Design to withstand normal, expected human behavior
- Design to combat lack of alertness
- Maintain active engagement in tasks
- Allow latitude in how tasks are accomplished.
- Avoid designs that require or encourage management by exception.

Matching Tasks to Human Characteristics (2)

- Distinguish between providing help and taking over.
  - Do not oversimplify the operator's task
- Maintain manual involvement or ways to update mental models.
- Design tasks to be stimulating and varied, to provide good feedback, and to require active involvement of the operators in most operations.
- Minimize activities requiring passive or repetitive action.

Designing to Reduce Common Human Errors



Reducing Human Errors

- Make safety-enhancing actions easy, natural, and difficult to omit or do wrong

  Stopping an unsafe action leaving an unsafe state should require one keystroke

- Make dangerous actions difficult or impossible

  Potentially dangerous commands should require one or more unique actions.

- Provide references for making decisions
- Integrate critical actions into one task.
- Follow human stereotypes and cultural norms

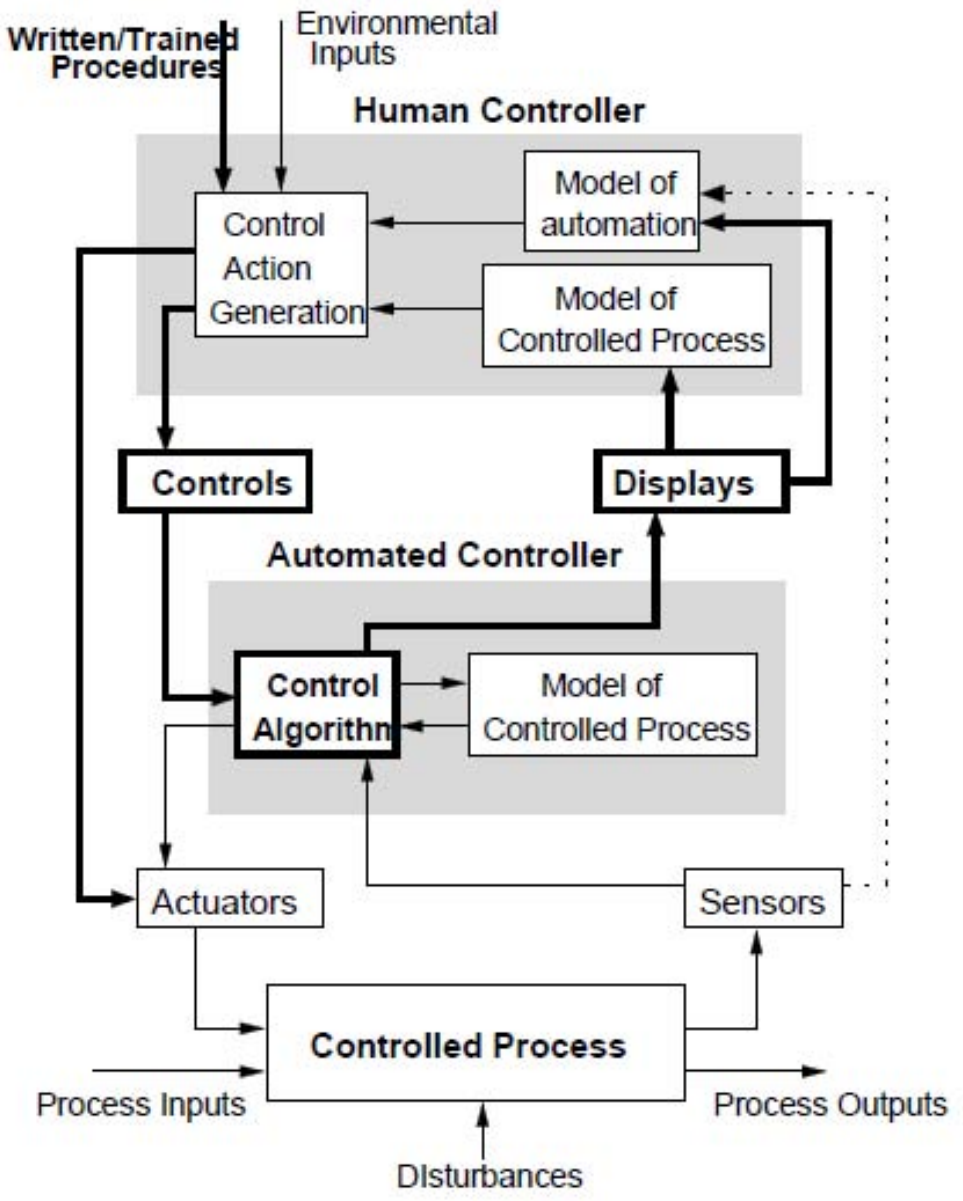
Reducing Human Errors (2)

- Make sequences dissimilar if need to avoid confusion between them
- Make errors physically impossible or obvious
- Use physical interlocks (but be careful about this)
- Make safety-critical steps incremental.
- Distinguish the override of safety-critical vs. non-safety critical errors or hazard indications.
- While safety interlocks are being overridden, their status should be displayed.
- After an emergency stop, require operator to go through entire restart sequence.

Reducing Human Errors (3)

- Distinguish processing from failure. Provide real-time indication that
    - Automated control system is functioning
    - Information about internal state (such as sensors and actuators), control actions, and assumptions about system state.
- Provide facilities for operators to experiment, to update their mental models, and to learn about system.
- Design to enhance operator's ability to make decisions and to intervene when required in emergencies.
- Allow operator to maintain manual involvement and to update mental models, maintain skills, and preserve self-confidence.

Support Maintaining Accurate Process Models



Inconsistent Behavior

- Harder for operator to learn how automation works
- Important because pilots (and others) report changing scanning behavior

Examples:

  - In go-around below 100 feet, pilots failed to anticipate and realize autothrust system did not arm when they selected TOGA power because it did so under all other circumstances where TOGA power is applied (found in simulator study of A320).
  - Similar thing happened in Cali accident
  - Bangalore (A320): A protection function is provided in all automation configurations except the ALTITUDE ACQUISITION mode in which autopilot was operating.

Unintended Side Effects

An action intended to have one effect has an additional one

Example (A320):

- Because approach is such a busy time and the automation requires so much heads down work, pilots often program the automation as soon as they are assigned a runway.

  In an A320 simulator study, discovered that pilots were not aware that entering a runway change AFTER entering the data for the assigned approach results in the deletion of all previously entered altitude and speed constraints even though they may still apply.

Modes

- Define mutually exclusive sets of automation behavior
- Used to determine how to interpret inputs or to define required controller outputs
- Four general types:
  - Controller operating modes (sets of related behaviors in controller, e.g., nominal behavior, shutdown, fault-handling)
  - Supervisory modes (who or what is controlling the component at any time)
  - Display modes (affects information provided on display and how user interprets it)
  - Controlled process modes

Mode Confusion

- Early automated systems had fairly small number of modes
  - Provided passive background on which operator would act by entering target data and requesting system operations
- Also had only one overall mode setting for each function performed
  - Indications of currently active mode and of transitions between modes could be decided to one location on display
- Consequences of breakdown in mode awareness fairly small
  - Operators seemed able to detect and recover from erroneous actions relatively quickly

Mode Confusion (2)

- Flexibility of advanced automation allows designers to develop more complicated, mode-rich systems
- Result is numerous mode indications spread over multiple displays, each containing just that portion of mode status data corresponding to a particular system or subsystem
- Designs also allow for interactions across modes
- Increased capabilities of automation create increased delays between user input and feedback about system behavior

Mode Confusion (3)

- These changes have led to:
  - Increased difficulty of error or failure detection and recovery
  - Challenges to human's ability to maintain awareness of:
    - Active modes
    - Armed modes
    - Interactions between environmental status and mode behavior
    - Interactions across modes
- Two types of problems
  - Interface interpretation errors
  - Indirect mode change errors

Interface Interpretation Errors

- Software interprets input wrong
- Multiple conditions mapped to same output

Mulhouse (A-320):

- Crew directed automated system to fly in TRACK/FLIGHT PATH mode, which is a combined mode related both to lateral (TRACK) and vertical (flight path angle) navigation.
- When they were given radar vectors by the air traffic controller, they may have switched from the TRACK to the HDG SEL mode to be able to enter the heading requested by the controller.
- However, pushing the button to change the lateral mode also automatically change the vertical mode from FLIGHT PATH ANGLE to VERTICAL SPEED, i.e., the mode switch button affects both lateral and vertical navigation.
- When the pilots subsequently entered “33” to select the desired flight path angle of 3.3 degrees, the automation interpreted their input as a desired vertical speed of 3300 ft. Pilots were not aware of active “interface mode” and failed to detect the problem.
- As a consequence of too steep a descent, the aircraft crashed into a mountain.

Interface Interpretation Errors (2)

Operating room medical device

- The device has two operating modes: warm-up and normal.
- It starts in warm-up mode whenever either of the two particular settings are adjusted by the operator (anesthesiologist).
- The meaning of alarm messages and the effects of controls are different in these two modes, but neither the current device operating mode nor a change in mode are indicated to the operator.
- In addition, four distinct alarm-triggering conditions are mapped onto two alarm messages so that the same message has different meanings depending on the operating mode.
- In order to understand what internal condition triggered the message, the operator must infer which malfunction is being indicated by the alarm.

Display modes:

In some devices user-entered target values are interpreted differently depending on the active display mode.

Indirect Mode Changes

- Automation changes mode without direct command
- Activating one mode can activate different modes depending on system status at time of manipulation

Bangalore (A320)

- Pilot put plane into OPEN DESCENT mode without realizing it. Resulted in aircraft speed being controlled by pitch rather than thrust, i.e., throttles went to idle.
- In that mode, automation ignores any preprogrammed altitude constraints. To maintain pilot-selected speed without power, automation had to use an excessive rate of descent, which led to crash short of the runway.

How could this happen?

Three different ways to activate OPEN descent mode:

1. Pull altitude knob after select lower altitude
2. Pull speed knob when aircraft in EXPEDITE mode.
3. Select a lower altitude while in ALTITUDE ACQUISITION mode.

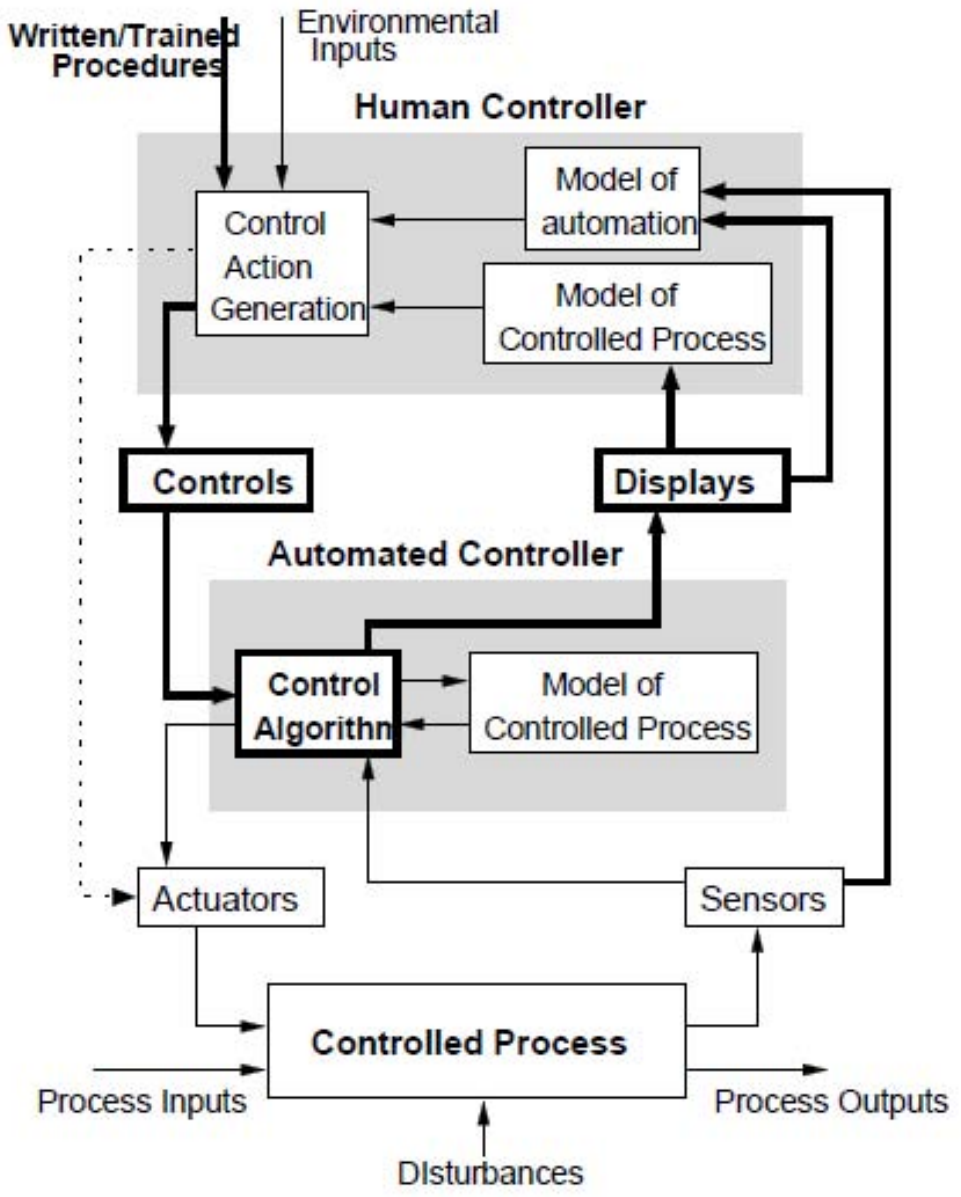
Indirect Mode Changes (2)

- Pilot must not have been aware that aircraft was within 200 feet of previously entered target altitude (which triggers ALTITUDE ACQUISITION mode).
- Thus may not have expected selection of lower altitude at that time to result in mode transition.
- So may not have closely monitored his mode annunciations.
- Discovered what happened at 10 secs before impact — too late to recover with engines at idle.

Coordination of Multiple Controller Process Models

- Crew resource management
- NW188 (John Thomas)

Providing Information and Feedback



Providing Information and Feedback

- Analyze task to determine what information is needed (STPA)
- Two types of feedback needed:
  - Effect of operator's actions
    - To detect human errors
  - State of controlled system
    - To update mental models
    - To detect system faults

Updating Process Models

- Automated control system should provide information about
  - Whether it is functioning (status indicator to distinguish between processing and failure)
  - Its internal state (such as state of sensors and actuators), its control actions, and its assumptions about the state of system.
- Provide for failure of computer displays (by alternate sources of information)
  - Instrumentation to deal with malfunction must not be disabled by the malfunction.
- Provide way for operators to test their hypotheses.
- Support detection of non-events

Updating Process Models (2)

- If operator must monitor computer decision-making, then computer must make decisions in a manner and rate operator can follow.
- Do not overload operator with too much information
- When task performance requires or implies need to assess timeliness of information, display should include time and date info associated with data
- Provide ways for operator to get additional information designer did not foresee would be needed in a particular situation.
- Provide alternative means for operators to check safety-critical information.

Updating Process Models (3)

- If important information changes in a very short interval before or after the operator issues a command (latency), make sure operator is aware of changes.
- Do not permit over-rides of potential safety-critical failures or clearing of status data until all data has been displayed and perhaps not until operator has acknowledged seeing it.
- While safety interlocks are being overridden, their status should be displayed. Design should require confirmation that interlocks have been restored before allowing resumption of normal operation.
- For robot systems,
  - Signal bystanders when machine is powered up
  - Provide warnings when hazardous zone is entered
  - Do not assume humans will not have to enter robot's area.

Detecting Faults and Failures

- Automated control system should provide information about
  - Whether it is functioning (status indicator to distinguish between processing and failure)
  - Its internal state (such as state of sensors and actuators), its control actions, and its assumptions about the state of system.
- Provide for failure of computer displays (by alternate sources of information)
  - Instrumentation to deal with malfunction must not be disabled by the malfunction.
- Provide feedback if commands are canceled (not executed) because of timeouts or other reasons.
- Operators cannot monitor performance if information not independent from thing being monitored.
- Inform operators about anomalies, actions taken, and current system state
- Fail obviously or make graceful degradation obvious to operator

Example Accident

Bangalore (A320):

- PF had disengaged his flight director during approach and was assuming PNF would do the same.
- Result would have been a mode configuration in which airspeed is automatically controlled by the autothrottle (the SPEED mode), which is the recommended procedure for the approach phase.
- However, the PNF never turned off his flight director, and the OPEN DESCENT mode became active when a lower altitude was selected.
- This indirect mode change led to the hazardous state and eventually the accident.
- But a complicating factor was that each pilot only received an indication of the status of his own flight director and not all the information necessary to determine whether the desired mode would be engaged.
- The lack of feedback or knowledge of the complete system state contributed to the pilots not detecting the unsafe state in time to correct it.

Alarms

- Issues
  - Overload
  - Incredulity response
  - Relying on as primary rather than backup (management by exception)

Alarms (2)

- Guidelines
  - Keep spurious alarms to a minimum
  - Provide checks to distinguish correct from faulty instruments
  - Provide checks on alarm system itself
  - Distinguish between routine and critical alarms. Form of alarm should indicate degree or urgency.
  - Indicate which condition is responsible for alarm
  - Provide temporal information about events and state changes
  - Require corrective action when necessary

Displaying Feedback to Human Controllers

- Highlight status of safety-critical components or variables and present complete state in unambiguous manner.
- Provide scannable displays that allow operators to monitor and diagnose using pattern recognition. Provide information, if appropriate, in a form in which patterns can be easily recognized.
- Make all information needed for a single decision process visible at same time.
- Avoid displaying absolute values. Show changes and use analog instead of digital displays when they are more appropriate. Provide references for judgment.

Displaying Feedback to Human Controllers

- Choose icons that are meaningful to users, not necessarily designers.
- Minimize semantic distance between interface displays and mental models (the form of the information needed by the user for processing).
- Design the control panel to mimic the physical layout of the plant or system.
- Flag rather than remove obsolete information from computer displays. Require operator to clear it explicitly or implicitly (e.g., let it scroll off top of screen).

Training and Maintaining Skills

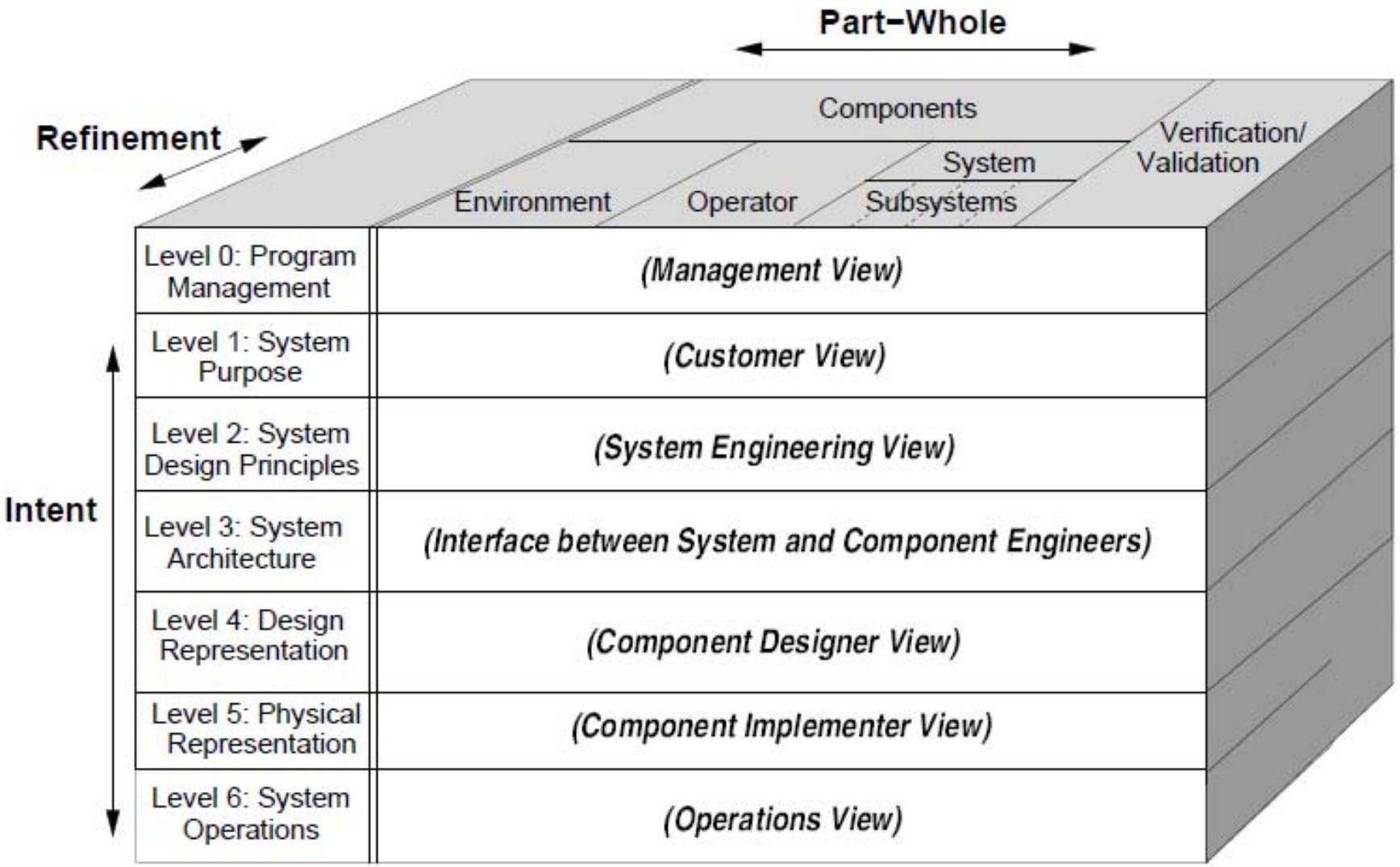
- May need to be more extensive and deep
  - Required skill levels go up (not down) with automation
- Teach how the software works
- Teach operators to think flexibly when solving problems
- Teach about safety features and design rationale,
  - Hazards and reason behind safety-critical procedures and operational rules
  - Potential result of removing or overriding controls, changing prescribed procedures, and inattention to safety-critical features and operations. Review past accidents and their causes.

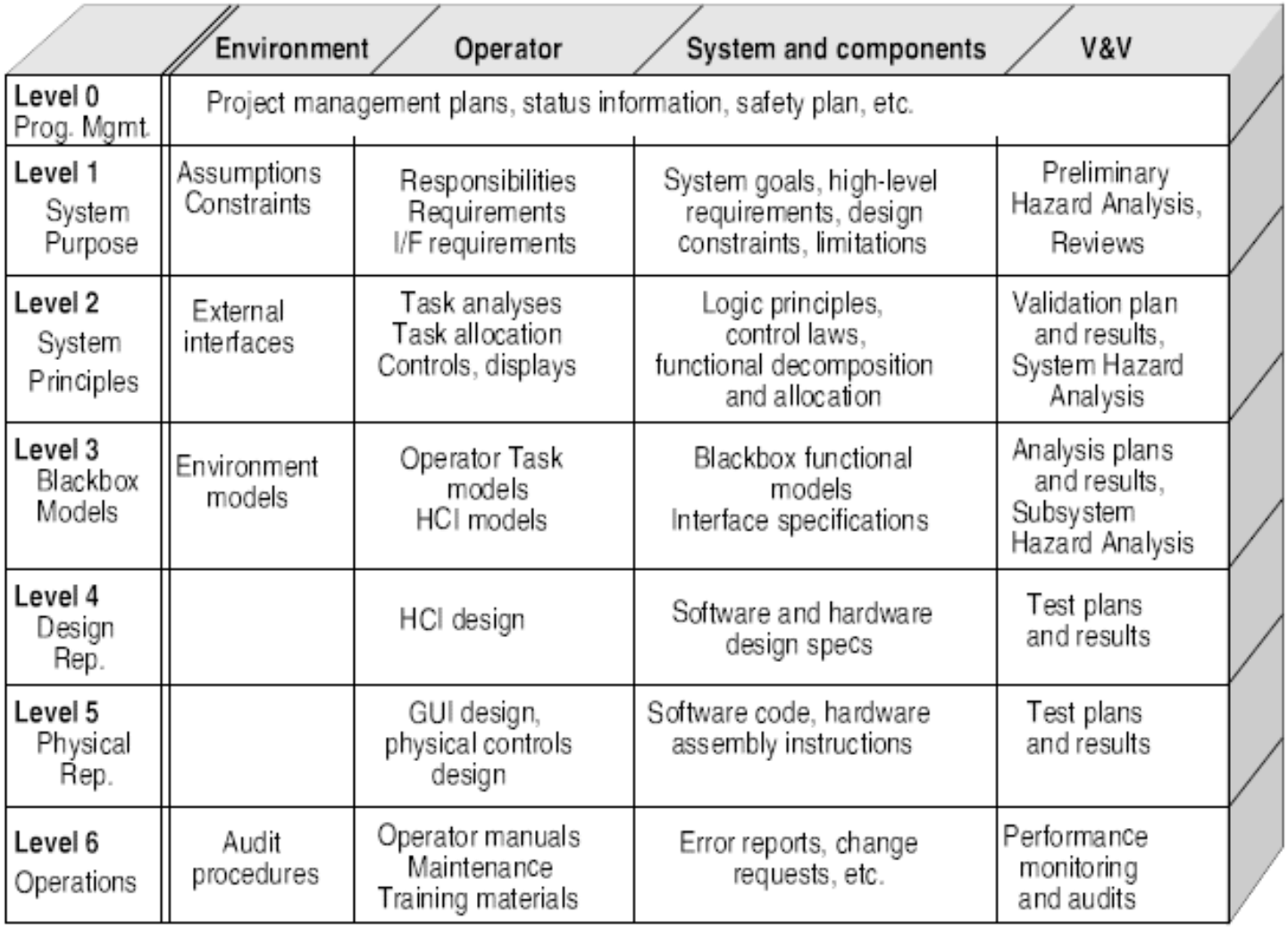
Training and Maintaining Skills

- Teach for general strategies rather than specific responses to develop skills for dealing with unanticipated events
- Provide in-depth understanding of process design
- Train operators to test hypotheses in appropriate ways.
- Train operators in different combinations of alerts and sequences of events, not just single events
- Allow for over-learning emergency procedures and for continued practice. Provide limits and specific actions to take in emergencies.
- Provide practice in problem solving.

Integrating Safety into System Engineering

- Start safety-guided design during concept formation
- Documentation is critical, particularly design rationale
- Intent specifications embed rationale in specifications
- Traceability


Part–Whole
Refinement
Components
Verification/
Validation
System
Environment
Operator
Subsystems
Level 0: Program Management
(Management View)
Level 1: System Purpose
(Customer View)
Level 2: System Design Principles
(System Engineering View)
Level 3: System Architecture
(Interface between System and Component Engineers)
Level 4: Design Representation
(Component Designer View)
Level 5: Physical Representation
(Component Implementer View)
Level 6: System Operations
(Operations View)
Intent



	Environment	Operator	System and components	V&V
Level 0 Prog. Mgmt.	Project management plans, status information, safety plan, etc.			
Level 1 System Purpose	Assumptions Constraints	Responsibilities Requirements I/F requirements	System goals, high-level requirements, design constraints, limitations	Preliminary Hazard Analysis, Reviews
Level 2 System Principles	External interfaces	Task analyses Task allocation Controls, displays	Logic principles, control laws, functional decomposition and allocation	Validation plan and results, System Hazard Analysis
Level 3 Blackbox Models	Environment models	Operator Task models HCI models	Blackbox functional models Interface specifications	Analysis plans and results, Subsystem Hazard Analysis
Level 4 Design Rep.		HCI design	Software and hardware design specs	Test plans and results
Level 5 Physical Rep.		GUI design, physical controls design	Software code, hardware assembly instructions	Test plans and results
Level 6 Operations	Audit procedures	Operator manuals Maintenance Training materials	Error reports, change requests, etc.	Performance monitoring and audits

16.863J / ESD.863J System Safety

Spring 2011