EVALUATING THE AUTONOMY OF UAVs

Herbert Hecht, Ph. D.
SoHaR Incorporated
Culver City, California
NO HUMAN PILOT

SAVES WEIGHT
SIMPLIFIES DESIGN
INCREASES LOSS
ACCEPTANCE
EXTENDS FLIGHT ENVELOPE

ALL HANDLING OF ANOMALOUS CONDITIONS MUST BE PROGRAMMED AND TESTED IN ADVANCE

EXCEPTION HANDLING
SOFTWARE TESTING

REQUIREMENTS BASED
- All stated requirements have been implemented
- Execution produces desired results

STRUCTURAL
- Traversal of implemented paths produces no undesirable results

Neither approach assures adequacy of exception handling
EXCEPTION HANDLING

VERY LITTLE LITERATURE

- EXCEPT FOR LANGUAGE CONSTRUCTS

NO GUIDANCE FOR SYSTEM LEVEL REQUIREMENTS FORMULATION

MOST SOFTWARE FAILURES IN WELL-TESTED SYSTEM ARE DUE TO FAULTY EXCEPTION HANDLING
MORE EXCEPTION HANDLING FAILURES

ALL FAILURES

GLOBAL FAILURES

“The main line software code usually does its job. Breakdowns typically occur when the software exception code does not properly handle abnormal input or environmental conditions – or when an interface does not respond in the anticipated or desired manner.”


“Therefore the identification and handling of the exceptional situations that might occur is often just as (un)reliable as human intuition.”

SPECIFYING EXCEPTION HANDLING IS DIFFICULT

EXCEPTION CONDITIONS ARISE FROM SEVERAL LEVELS
SPECIFYING EXCEPTION HANDLING IS DIFFICULT

EXCEPTION CONDITIONS ARISE FROM SEVERAL LEVELS

EXCEPTION CONDITIONS ARE MORE DIFFICULT TO UNDERSTAND THAN MAIN LINE REQUIREMENTS
SPECIFYING EXCEPTION HANDLING IS DIFFICULT

- Exception conditions arise from several levels.
- Exception conditions are more difficult to understand than main line requirements.
- Exceptions occur infrequently but require disproportionate effort.
SOURCES OF EXCEPTIONS

OPERATIONAL REQUIREMENTS
  LOSS OF PROPULSION, ELECTRIC POWER, COMMUNICATION, THERMAL CONTROL

IMPLEMENTATION DETAIL
  CALIBRATION ANOMALIES, ACTUATOR STATES, SENSOR INPUT

COMPUTING ENVIRONMENT
  HARDWARE FAILURES, MEMORY ERRORS, EXECUTIVE, MIDDLEWARE

MONITORING AND SELF-TEST
  OVER-TEMPERATURE SENSORS, SYSTEM PERFORMANCE TEST

APPLICATION SOFTWARE
  ASSERTIONS, VIOLATION OF TIMING CONSTRAINTS, MODE CHANGES
WHO IS RESPONSIBLE?

OPERATIONAL REQUIREMENTS

IMPLEMENTATION DETAILS

COMPUTING ENVIRONMENT

MONITORING AND SELF-TEST

APPLICATION SOFTWARE
REQUIREMENT GENERATION

OBJECTIVE
- EXCEPTION CONDITION AND ACTION

ALGORITHM
- QUANTITATIVE CONDITION DESCRIPTION
- TIMING AND RESPONSIBILITY FOR ACTION

ASSIGNMENT
- SPECIFY SOFTWARE IMPLEMENTATION OF ALGORITHM
DOES IT ADD UP?

CONCEPT

SYST. REQ'MTS

SOFTWARE REQ'MTS

SOFTWARE DESIGN

CODING

OBJECTIVE

ALGORITHM

ASSIGNM'T

OPERATIONAL REQ'M'TS

IMPLEMENTATION

COMPUTING ENV.

MONIT. & SELF-TEST

APPLICATION SOFTWARE

CONCEPT

SYST. REQ'MTS

SOFTWARE REQ'MTS

SOFTWARE DESIGN

CODING
SOLUTIONS TO THE PROBLEM

- SHARING EXISTING PRACTICES
- SHARING EXPERIENCE
- CREATING AND SHARING TOOLS

- INTEREST GROUP
- STANDARDS WORKING GROUP
- RECOMMENDED PRACTICE
HERBERT HECHT
herb@sohar.com
310.338.0990 X110