

Generation of Ada Code from Specifications for NH90 Computers

Gerd Budich, NH90 Project Manager for Software Development
Eurocopter Deutschland GmbH
Munich, Germany

Abstract

Eurocopter Deutschland develops in the NH90 Programme the software for three computers

- Core Management Controller (CMC)
- Display and Keyboard Unit (DKU)
- Mission Tactical Computer (MTC)

By setting up software development for the three computers quantity problems had to be solved, which lead otherwise to large error-prone SW code:

- CMC and MTC are both Milbus Controller for the Core- or Mission- Bus and have to handle between 5,000 and 10,000 different data;
- DKU is designed to work with up to 1,000 pages, each page consisting of 14 rows of 29 characters.

The software for all three computers shall be designed for a large number of NH90 variants, with different sets of equipment.

Previous experiences (from the TIGER Programme) showed that software for data of that amount and variety cannot be hand-coded with the required quality and efficiency.

Therefore decisions have been taken to produce a suite of Code Generators, which uses extracts of the specification tools as inputs and produces Ada source code which were directly embedded into the hand-coded software programmes.

The presentation gives an overview about the developed tool chain, and discusses the problems of

- verification of the specifications
- verification of software code generators output
- configuration management
- software qualification

and compares the efficiency of generated and hand-coded software.

Introduction

The three Computers – developed by ECD - are embedded in the NH90 Avionics system, which consists of three major Subsystems:

- the Flight Control System
- the Core System
- Mission System

In Figure 1 the Flight Control System and the Core System are combined to the Basic System. The Mission System is represented by the Mission equipment of the TTH.

Both Core and Mission System are designed around a dual redundant MIL Bus according to STANAG 3838. The CMCs build a dual redundant pair of Bus Controllers for the Core Bus, while the MTC plays the same role for the Mission Bus. The MTC is the bridge for data transfers between Core and Mission equipment.

Other components of the Core System are the Navigation System, the Plant Management Computers (PMC), the Communication and Identification System and the Core Data Transfer Device.

The components of the TTH Mission are Weather Radar, Mission Data Transfer Device, Forward Looking Infrared, Helmet Mounted Sight/Display, Tactical Communication, Electronic Warfare System, Obstacle Warning System and Digital Map Generator.

The main user interface for both Core and Mission System are up to 5 8" x 8" Multi Function colour Displays (MFDs) and up to 4 Display and Keyboard Units (DKUs).

Currently 11 NH90 Helicopter Variants are contracted, which differ from the needs of four countries with 9 users (Army, Air Force, Navy). Solutions for other countries which are interested in the NH90 helicopter have to be proposed.

The helicopter variants configurations shall be as flexible as possible to allow the users to change the Avionics System as maintenance task with a minimum of re-qualification.

Due to there role as MIL Bus controllers CMC and MTC have to be aware of all equipment which are connected to Core or Mission Bus and of the definition of all frames, messages and data being exchanged between the equipment via the MIL Busses.

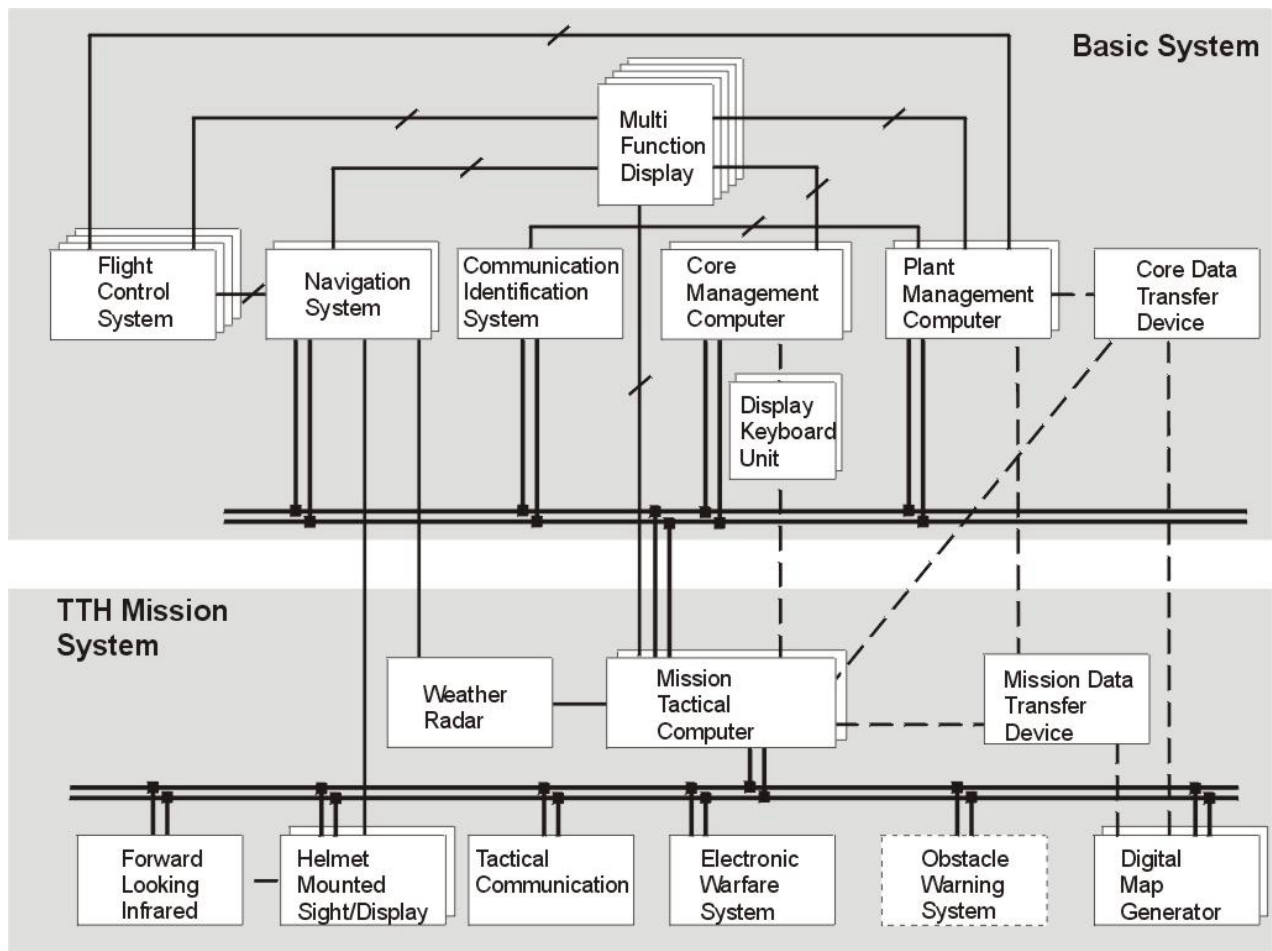


Figure 1: Basic System and TTH Mission System (video connections not shown)

Additionally both Computers are responsible to

- command the equipment,
- monitor the status of actually connected equipment,
- provide information about the avionics system configurations
- provide the data for the DKUs and control their pages and the availability of commands and display parameters (CMC for Core pages and MTC for Mission Pages).

This causes the general requirements that

- both computers CMC and MTC shall be able to handle the amount of up to 10,000 data each
- the DKU had to be designed to handle up to 1,000 operational and maintenance pages.

All three computers shall support the changes of configurations with minimum software development and software qualification effort.

NH90 Software Development Process

To support software maintenance during the In Service phase the participating companies defined the rules and procedures for the software development in a Software Development Handbook

(SDHB), which was harmonised with the customer and which builds the frame for the development of the computers CMC, DKU, MTC (TTH and NFH) and PMC by:

- adopting the Standards DOD STD 2167A and RTCA DO178A to an
- incremental Software Development Process
- with different Software Variants,
- four Software Criticality Levels (1, 2A, 2, 3) using the four Levels (A,B,C,D) of RTCA DO178B as Guidelines,
- software documentation according to the Data Item Descriptions of DOD STD 2167A.

Software Requirements Analysis

To support the definition of different software variants (for the NH90 helicopter variants) the software requirements for CMC and MTC are organised in volumes, each related to a subsystem or equipment defining the operational and tactical functions.

An additional volume – common for CMC and MTC - defines the requirements for an Avionics Operating System (NH90 System Software: NSS)

comprising all software requirements not related to operational and tactical functions.

Requirements are defined mainly textually using the tool Statemate and the advantages of state charts to specify or illustrate the behaviour of the functions.

The Interface Data for the whole Core and TTH Mission System are defined in a common Avionics Data Base System (ADBS) which is used, not only

- to derive the Software Interface Requirements Specifications (IRS), but also
- for set-ups of test data for the test systems of Software Test Benches, RIG-, Ground- and Flight Testing.

The approach for the DKU was based on Tiger experiences, showing that the page definitions are subject for permanent changes during the harmonisation process of the MMI of the Helicopter between the customer and industry and in In Service phase too. Therefore:

- Independence of page definitions and software development of DKU functions was highly recommended.
- Simulation of DKU pages had to be provided for the definition process on host computers.
- Short turn around times between page definition and simulation of pages on host or testing on target had to be ensured.

To solve these requirements the following derived requirements were established:

- the operations of the DKU pages, their elements, organisation and colours have to be defined in a "DKU Control and Moding" document and harmonised between customer and industry;
- a database tool (DUET: DKU Unified Engineering Tool) shall be developed to support consistency among page definitions, DKU software and CMC / MTC software;
- the DKU software shall be separated into
 - a Page Interpreter realising all functions of the "DKU Control and Moding", the handling of screen and keyboard and the interface to the CMCs and MTCs and into
 - Tables representing the pages, their data and actions

Page Interpreter and Tables should be separately loadable software units;

- to ensure fast turn around times the software code for the Tables should be generated out of the DUET database.

Design Concept for CMC and MTC Software

CMC and MTC have a high degree of commonality:

- both are bus controllers for the Core or Mission System,
- the redundant computer is Remote Terminal on the bus,
- both computers have the same type of additional interfaces: Arinc lines, RS485, RS232 and discrete lines
- both have the same software criticality level according to the NH90 hazard analysis.

The formats and types of data in ADBS or DUET defines how the data are exchanged between equipment physically. For software generation these definitions have to be transferred into Ada specific variables and types.

To minimise software development and software qualification effort for different helicopter variants the following decisions have been taken:

- establish a software development process for the generic common Avionics Operation System (NH90 System Software: NSS) in Ada with all activities defined in the SDHB for a software product;
- develop a suite of software generators for those NSS components which have to be adapted to build a specific software variant for CMC or MTC to covers the needs of one or preferable several helicopter variants;
- develop an additional database tool for the software specific aspects to link all external data to software variables and types and to provide all necessary data to support the generation process. This tool is named: Operational Data and Interfaces in NH90: ODIN;
- qualify this NSS and the adaptation process (this means the transfer of data from the specification data bases into ODIN and the transfer from ODIN into Ada code using the generators) by a specific Evaluation Project which is representative for all used cases and classes (in terms of quality not quantity).

Figure 2 shows the components of the NSS and their adaptation to actual CMC or MTC software variants by introducing CMC or MTC specific Ada Code.

The main components and their adaptation are:

- System Controller
adapted: call of operational and tactical functions by tasks of different frequency
- the Onboard Processing Software
no adaptation

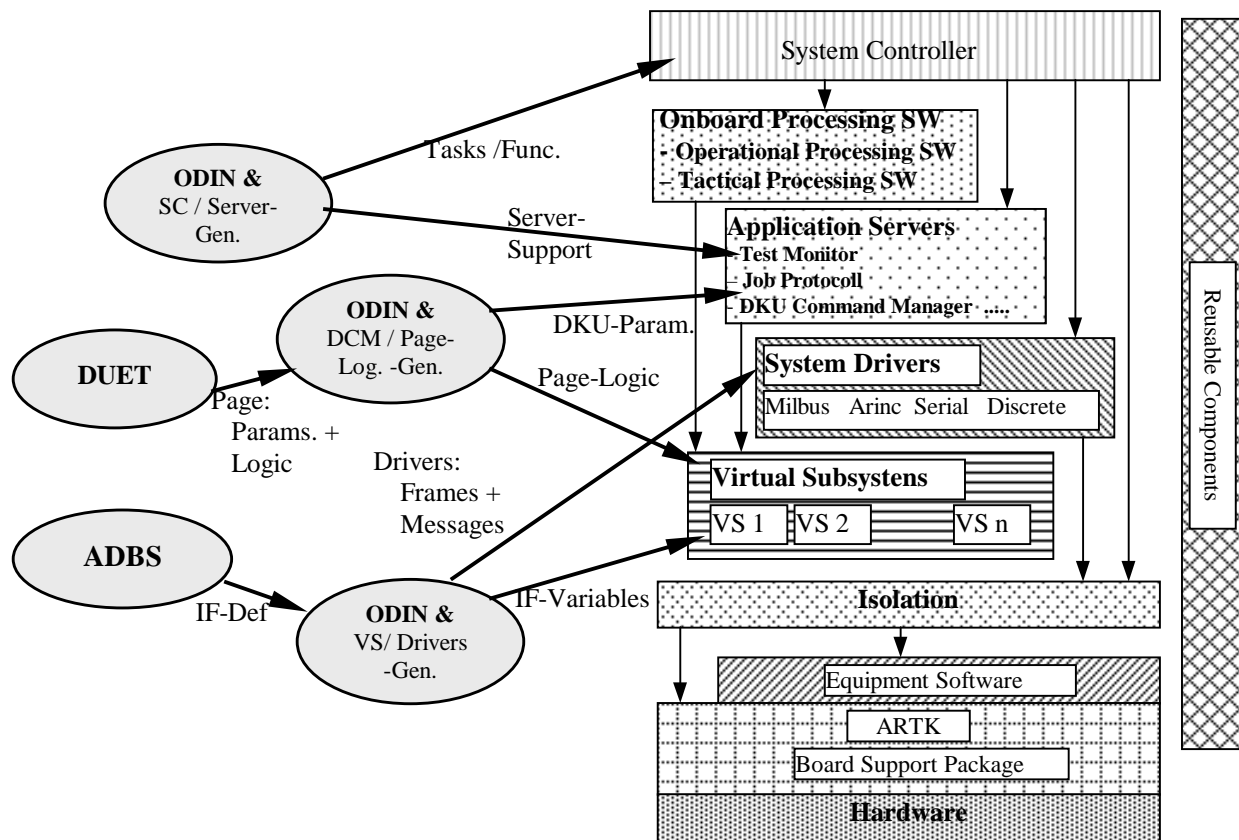


Figure 2: NSS Components and their Adaptation

- Application Servers:
 - Test-monitor – not adapted,
 - Job Protocoll – adapted
 - DKU Command Manager- adapted:
 - Non Volatile Memory – adapted
 - Common Data Base – adapted
 - List Server not adapted
 - Shared Memory Server adapted
- System Drivers: - adapted:
 - structure of all frames and messages for MIL Bus and Arinc interfaces,
 - structure of the data transferred via RS485.
- Virtual Subsystems – adapted:
 - all interface data exchanged between CMC or MTC and the equipment with conversion into Ada variables and types if they are of interest for these computers.
 - page logic which defines the status of DKU display and command parameters depending on the availability of the related equipment.
- Isolation provides independence for the NSS from the layers underneath platform (host or target) no adaptation.

are of textual form. They have to be designed, coded, unit tested and integrated by a normal software development process as defined in the SDHB.

For the generation of the adaptation Ada code data of the databases of the Software Requirements Analysis

- ADBS for the interface data and
- DUET for the page- layouts, -parameters and -logic.

are transferred into the ODIN Database, and linked to the software specific variables and types used by the Onboard Processig Software.

When all data relevant for the software variant are linked the generator suite can be started, which generates the Ada code and stores it in the required directories of the Software Development Environment – based on Rational Apex.

The check if all components fits together is made by the Ada compiler. No errors means that the interfaces of generated and hand written code are consistent – the first important step in software integration.

The requirements for the **Onboard Processing Software** which controls the different equipment

Design Concept of the DKU Software

To be compliant to the DKU requirements mentioned in chapter Software Requirements Analysis the development process of figure 3 was chosen:

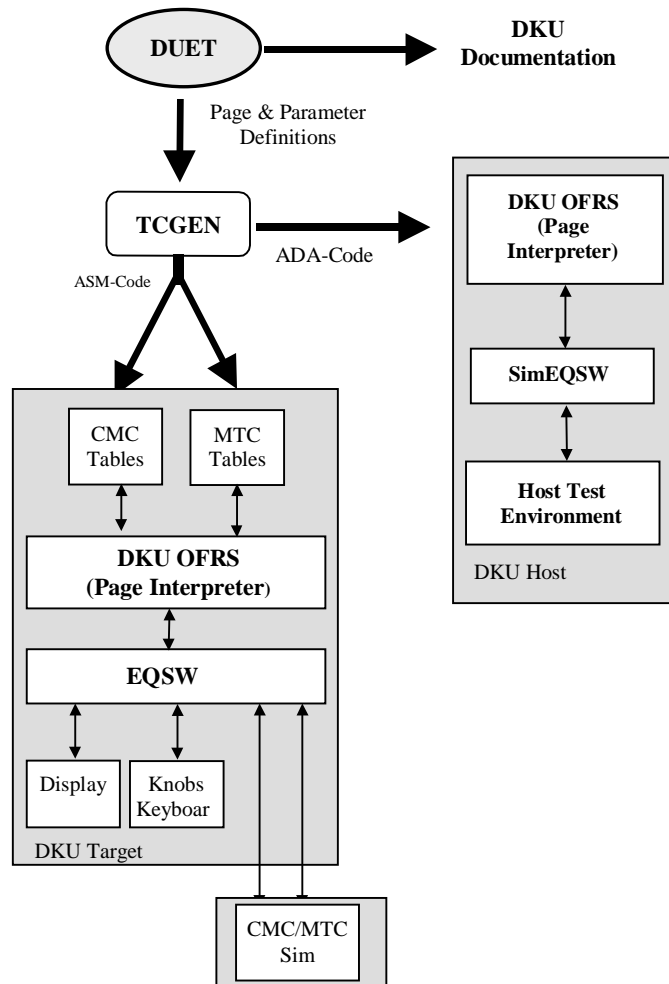


Figure 3: DKU Components and Table Generation

- develop an Oracle based tool – DUET - which
 - supports the definition of DKU Pages according to the rules of the DKU Control and Moding document;
 - provides documentation of these page definitions for the users of the helicopter;
 - supports the definition of data for the interfaces between DKU and CMCs / MTCs;
 - supports the definition of the logic for the availability of display and command parameters depending e.g. on the status of the installed equipment (this logic has to be generated for and integrated in CMCs and MTCs);

- supports the definitions of characters, fonts and line styles displayed on the DKU screen;
- ensures configuration control for the development of several DKU software variants
- develop the Page Interpreter (the DKU OFRS - Onboard Flight Resident Software) with all activities defined in the SDHB for a software product in Ada to be used
 - on target and
 - in a simulation environment on host
- develop a Table and Code Generator (TC Gen) to generate Ada code which represents the pages to control the equipment of the Core system (CMC tables) and of the Mission system (MTC tables)
- define a set of qualification pages which uses all functions defined in the DKU Control and Moding to qualify
 - the functionality of the DKU page interpreter and
 - the generation process from DUET definitions via Table and Code Generator to the software code for the DKU on target and host.
- develop a program to provide stimulation for the DKU parameters for the qualification pages (CMC/ MTC Simulation).

Verification of the Specifications

As shown in the previous chapters code-generation is performed not for functional requirements but for data definitions of several structures and for

the access routines to those data. The specification tools are

- ADBS for interface data of Core and Mission systems and
- DUET for Core and Mission pages and the processing of commands and parameters in CMC/MTC (DKU Command Manager)
- DUET for software specific parts as System Controller, Test Monitor

A prerequisite for the generation of code from specification is the high quality of the specified items.

Within NH90 System and software development no formal verification of specifications have been chosen. The quality improvements of the content of the

databases rely on the development process and the fact that all users of those data work with the same database and report inconsistencies which are traced by a Problem Reporting tool:

- Equipment supplier of COTS equipment deliver their interface data (Interface Design Document: IDD) in electronic form, the data are checked by the entry programs of the database for consistency with already existing data and types;
- For newly developed equipment (e.g. the MFDs): after manual data entry with consistency checks the source for the Interface Design Document is ADBS;
- During Software Requirements Analysis, the data of ADBS are permanently harmonised with the data dictionary of the specification tool Statemate to ensure consistency between Software Requirements Specification (Statemate) and Interface Requirements Specification (ADBS);
- during the transfer of data from ADBS to ODIN the data are checked for consistency with already existing definitions of earlier software versions for both CMC and MTC and the related software types;
- Testers on Software Test Benches and Rigs simulate missing equipment or monitor the interface data with the test system Anais; the source for the set-up of the interface data definitions for Anais is ADBS;
- the data definitions for the Flight Test Instrumentation equipment are derived from ADBS too.

For the quality improvement of DUET for DKU definitions the approach that all participants uses the same database is chosen too:

- during the definition of the pages
 - DUET provides support for consistency with the rules of the DKU Control and Moding,
 - the layout and some functionality can be checked directly by the simulation of the DKU on host (see figure 3)
- these definitions – and the re-hosted DKU Page Interpreter - are used in the cockpit simulators to harmonise the MMI concepts with the users – changes/improvements will be incorporated into DUET before the next evaluation loop;
- the documentation for this harmonisation and later for the DKU specification is generated with DUET;
- during Software Requirements Analysis the interface data for the pages are checked with the data dictionary of Statemate for consistency;

- finally the transfer of interface definitions and page-logic being handled by CMCs and MTCs evaluates the consistency with related software types.

Verification of Code Generator Outputs

Both the NSS and the DKU Page Interpreter software are in their basic functionality hand written software according to the development requirements of the SDHB specified and implemented.

To verify their functionality the generated parts have to be added:

- for NSS: the adaptation has to be performed
- for DKU: page tables have to be generated and loaded.

While Ada code generation is designed to cope with e.g. the mass of interface data (up to 10,000 for CMC and MTC each) and DKU pages (up to 1,000) the verification of the generation processes has to work with a limited number of items.

Two evaluation projects have been established to qualify:

- the NSS and the adaptation of the NSS, and
- the DKU Page Interpreter and the generation of tables.

Evaluation Project for NSS

The development process described in Figure 2 is used for the NSS evaluation project, but specific evaluation data have been specified. The possible different outputs of the generators have been analysed and grouped into classes. Requirement is that items of each class have to be treated at least once.

Currently are defined e.g.:

- for System Controller: 30 operations to be called in 20 Tasks on the two boards of CMC and MTC as well as 120 events (of 30 classes) to be handled in one control table
- for Onboard Processing Software: one operational function dealing with all items and invoking all Application Services
- for DKU Command Manager 10 DKU pages
- for System Drivers:
 - 350 signals and 100 messages in 4 frames for MIL Bus
 - 300 signals and 100 labels in 2 frames for Arinc
- for Virtual Subsystems:
 - 100 different data structures (for the Interface Data)
 - 20 parameters to test the page logic
- a.s.o.

For each class at least one generated item will be handled like hand written Ada code: design, code and unit-testing have to be documented and inspections have to be performed.

For the Evaluation project all activities of Formal Qualification Testing have to be performed with all test documentation.

Evaluation Project for DKU

The development process of figure 3 is used: For the evaluation of the DKU Page Interpreter and the Table Generation about 30 specific evaluation pages have been defined (CMC and MTC tables) dealing with all rules specified in the DKU Control and Moding document.

The CMC/MTC simulator has been adapted to react properly for the 30 evaluation pages. The Page Specification Document is generated from the DUET database and inspected.

To automate the fixed sequences of data entries on the keyboard - for each of the 30 evaluation pages - a Test Card Interpreter has been develop, which interprets commands and transfers the equivalents in form of key strokes to the DKU.

To support regression testing the output of the Page Interpreter to the DKU screen and the transfer of data between Page Interpreter and CMC/MTC simulation will be logged in files.

Configuration Management

The tools to support Software Requirements Analysis (Statemate, ADBS, DUET) have their own configuration management features, ensuring version control, restricted access rights to ensure data integrity and security. Here the aspects of change management after a Software Specification Review (SSR) will be discussed.

If the SSR is performed successfully the Software Requirements Specification and the Interface Requirements Specification will be frozen to build the Allocated Baseline. Detected problems will be described in Problem Reports (PRs), solutions will be specified in Engineering Change Requests (ECRs), if software is affected SW Change Notes (SWCNs) will be attached. A strict tracing of PRs and ECRs is of crucial importance for the project.

Due to the high degree of generated software the term "SW is affected" has become a new quality. The most important issue for the software team is the analysis result that "hand written SW is affected". In all other cases the generative approach

will incorporate the changes automatically when a new software release is build.

To this fact the software team became much more tolerant to those changes, which can be incorporated on a short notice.

The really important issue of Configuration Management is that the software release building is done based on the same frozen snapshot of the specification databases.

Software Qualification

Due to the Evaluation Projects a large amount of software is qualified independent from the actual software variants for the helicopter variants.

What has to be demonstrated for each software variant is that all generated classes of items are covered in the Evaluation Project.

Formal Qualification Testing of software variants can concentrate on Onboard Processing Software which controls the equipment or supports the pilots in their mission activities.

Efficiency of Code Generation Processes

The Code Generator Suite for the adaptation of the NSS for CMC or MTC needs comprises

- 40,000 lines of Perl code.
- to generate for the current software versions of
- CMC: about 280,000 lines of Ada code (main drivers are about 5000 interface data) and of
 - MTC: about 240,000 lines of Ada code.

This is approximately 60 % of the total amount of the embedded software.

For the DKU the requirement to be able to handle up to 1,000 pages and react on MMI changes flexible and in acceptable time was the basis for the chosen development approach. Early Tiger experiences showed that for an average page about 400 lines of hand written Ada code was necessary.

The current DKU Table and Code Generator comprises less than 10,000 lines of Perl code.

Conclusion

The development concept to generate software out of specification databases is one of the most important backbone to develop software in teams with limited manpower, to react on changes flexible and in short turn around times and finally to be able to cope with the number of helicopter variants of the NH90 program.