# Data Fusion Development Concepts within complex Surveillance Systems


**Felix Opitz** **Werner Henrich** **Thomas Kausch**

Defence and Communications Systems/Air and Naval Defence
EADS Deutschland GmbH
Woerthstr. 85, D-89077 Ulm
Germany

felix.opitz@sysde.eads.net werner.henrich@sysde.eads.net thomas.kausch@sysde.eads.net



**Abstract –** *Today surveillance systems are applied in many civil and defence areas and contain data fusion systems as embedded software items. Generally these systems are developed in accordance with different industry or governmental standards. Based on the experience the authors made within the development of several data fusion projects, this paper shows, how the software development standards can be tailored for a specific embedded data fusion process. It serves as a description of industry practice; identifies problems in applying software standards on the information fusion process, and demonstrates how existing standards need to be complemented/extended to include the area of information fusion.*



**Keywords:** data fusion, sensor data fusion, software development, integration & test, air traffic control, air defence system.


## 1 Introduction

Data Fusion is applied in modern surveillance systems like air traffic or satellite orbit control, coastal surveillance, air defence systems or more generally combat or battle management systems. These systems are commonly developed in accordance with different industrial and governmental standards, which must therefore also be applied to the development of the embedded data fusion. This is motivated by different aspects: The surveillance systems are of a complexity and size not manageable without a coordinated development process. One has to work with multiple developers simultaneously on the embedded software items like data fusion and resource management and the hardware items, e.g. sensors and effectors, over extended time. Therefore one has to deal with simultaneous engineering processes, which can only be managed by the application of formal development standards. The idea of the development standards is to support the quality of a software product by optimising the software engineering process. This is especially important for above mentioned surveillance systems. Because of their safety criticality the developed system must satisfy a high level of quality.

The commonly accepted international standard is the so called ISO/IEC 12207 standard [1]. It was prepared by a joint technical committee of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). NATO projects use the AQAP-160 standard, which is a modification of the above ISO/IEC 12207 standard. For military projects the DoD-STD-2167A was widely used [2]. The V-Model is popular for German government contracts and is highly applicable for the development of complex embedded systems [3, 4]. The idea of these standards is rather similar.

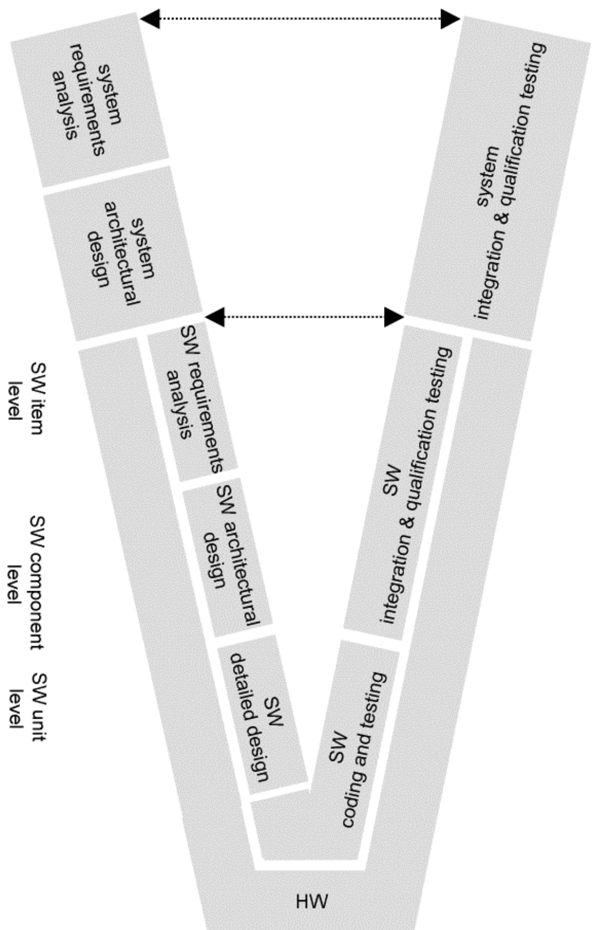


Fig. 1. The development model (simplified).

The development process is partitioned into the following steps (according to ISO/IEC 12207):

- System requirements analysis & design
- Software requirements analysis
- Software architectural design
- Software detailed design
- Software coding and testing
- Software integration & qualification testing
- System integration & qualification testing

For each step one has to ensure the traceability of requirements, consistency, test coverage, appropriateness, conformance, and feasibility.

## 2 Data Fusion Development

However the formal standards define only an abstract scheme for the data fusion software engineering process. It is important to tailor these schemes with the scientific and engineering background, which underlies the specifics of data fusion. The perhaps most important difficulty is the dependency between

- data fusion,
- sensor suite,
- targets, and
- environment.

The sensor performance and therefore the data fusion is influenced by the effects of the natural environment and the considered targets. The requirements analysis and design phases must take into account these dependencies, therefore a deep knowledge in sensor systems is absolutely necessary. One needs concepts for the test and integration activities. Whereas the data fusion development up to the software integration and test phase is performed in a simulated environment, the system integration & test uses real entities. Therefore, one needs concepts to handle the gap between

- simulation and
- reality [5].

In the following we discuss the specifics of data fusion development concepts related with the general development standards. Finally, data fusion is highly accessible to modularisation [6], an architectural aspect which is supported by the usage of coordinated development based on standards. This increases the reusability and extendibility and has impacts on economic aspects.

### 2.1 System requirements analysis & design

Within the system requirements analysis the intended use of a system is analysed and documented. This is done by requirements, which define e.g.: the functions and capabilities, operational and user aspects, safety and security topics, interfaces, maintenance, and design constraint. The succeeding system design establishes a top-level architecture of the system through the identification of software and hardware items and manual operations [1].

The system requirements and design aspects of a surveillance system connects the physical, information and cognitive domain [7] by determining the flows between these items (fig. 2).

#### *2.1.1 Physical domain: Hardware items*

The physical domain, which is relevant for data fusion, consists of the sensor and effectors suite. The sensor suite may consist of primary and secondary radars, electro-optical sensors (IR/laser), ESM or acoustic sensors. The effector suite depends on the system objectives. It may contain communication, control or warning systems, missile and gun systems, decoys, or electronic warfare segments.

#### *2.1.2 Information domain: Software items*

This is the domain of the data fusion, the resource management and the HMI. In open structure systems these components are allocated to an application layer and are connected and controlled via middleware.

#### *2.1.3 Cognitive domain: The human user*

Finally one has also to embed the future user, which is the final decision maker. Therefore an important point is to design the interactive system in such a way, that raw data, documents, business and operational models, etc. are transformed into a form, which is intuitively usable by the user, for his decision making process.

#### *2.1.4 Flows*

Finally it is the task of the system design to define the relevant flows, which connect the human user with the sensors and effectors via software items like data fusion or resource management depending on the different system missions.

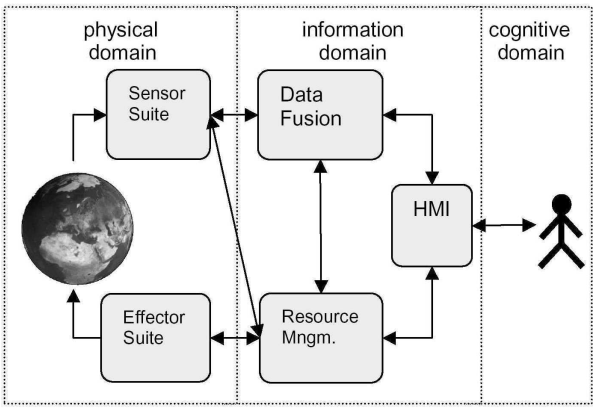


Fig. 2. The domains and the flows.

### 2.2 Software requirements analysis

During Software requirements analysis for each software item the requirements are derived from the system requirements [1].

Therefore the requirements for the data fusion (item) are derived, taking into account the system requirements defined by the flows between hardware and software items as well as the human operator. To do so, one should start by realizing the constraints of the possible input data.

### 2.2.1 *Constraints*

Normally one can divide the data provided by the sensor suite into three topics:

- Kinematics
- Attribute information
- Sensor internal information

The kinematics describe the position (velocity) state and accuracy of the targets for a determined timestamp. This may be done by delivering measurements or prefiltered information (tracks). Several sensors deliver also additional data related to the target attributes, interesting for classification, identification and discrimination purposes, or data about the sensor internal status like time energy budget, mode, turn rate etc.

The performance of these data has to be known with respect to quality and quantity (fig. 3): Quality is given by the accuracy of the measurements expressed by its standard deviation. Resolution is the sensor capability to distinguish two neighbour targets. Accuracy and resolution normally depend on the sensor band-width, beam-width, noise und pulse shape. Other factors are confidence and possible ambiguities of sensor data. Range and range rate measurements of radars may be ambiguous due to the choice of the pulse repetition frequency of the radar system. A low pulse repetition frequency enables unambiguous range measurements but suffers from ambiguous Doppler results whereas a high pulse repetition frequency produces unique range rate measurement for the price of ambiguous range measurements [8, 9]. Sensor level data processing like signal extraction and filtering implies constraints with respect to the detected target spectrum and manoeuvrability concerning to linear and cross acceleration. Furthermore one should take into account the spatial coverage of the sensor suite. Quantitative sensor constraints may depend on the scanning behaviour of the involved sensors. Some sensors still are based on mechanically scanned antennas (MSA) whereas more advanced sensors use the benefit of electronically scanned antennas (ESA). Mechanically scanned antennas perform sampling by rotating the antenna and posses a fixed update rate, whereas the variable update rate of electronically scanned antennas may depend on the environment around the target and the target instantaneous manoeuvres. Caused by the scanning behaviour in conjunction with the sensor internal tracking system, the measurements may be delivered to the data fusion system with different delays. Overall sensor constraints are measurement rates and total number of tracks. Besides those sensor specific constraints the data fusion system depends on hardware or software interfaces and the capabilities measured in MIPS, MEMS or BAUDS, or standards to ensure interoperability (e.g. STANAGs).

### 2.2.2 *Functional requirements*

To describe the functionalities of a data fusion the so called JDL model is generally applied (fig. 4). Level 0 fusions or sub “object assessment” is related to the sensor hardware environment. It deals with extraction and closely hardware related signal processing. Therefore it is performed within the sensor itself. Level 1 considers the object assessment The overall aim of this process level is to find a unique representation of all objects in the environment considered. Therefore the real objects within the surveillance area are described by so called tracks, which are built by data association and state estimation techniques. The 2nd and 3rd level deals with situation and impact assessment. Within these parts the relationships between several objects are studied. This may be of different types, e.g. spatial-temporal, organizational, causal, and similarity. The process refinement in level 4 tries to optimise the ongoing fusion process itself. This may be executed e.g. by adaptive data acquisition and processing. Through a 5th level called “cognitive refinement” and a human machine interface the result of data fusion is presented to the operator.

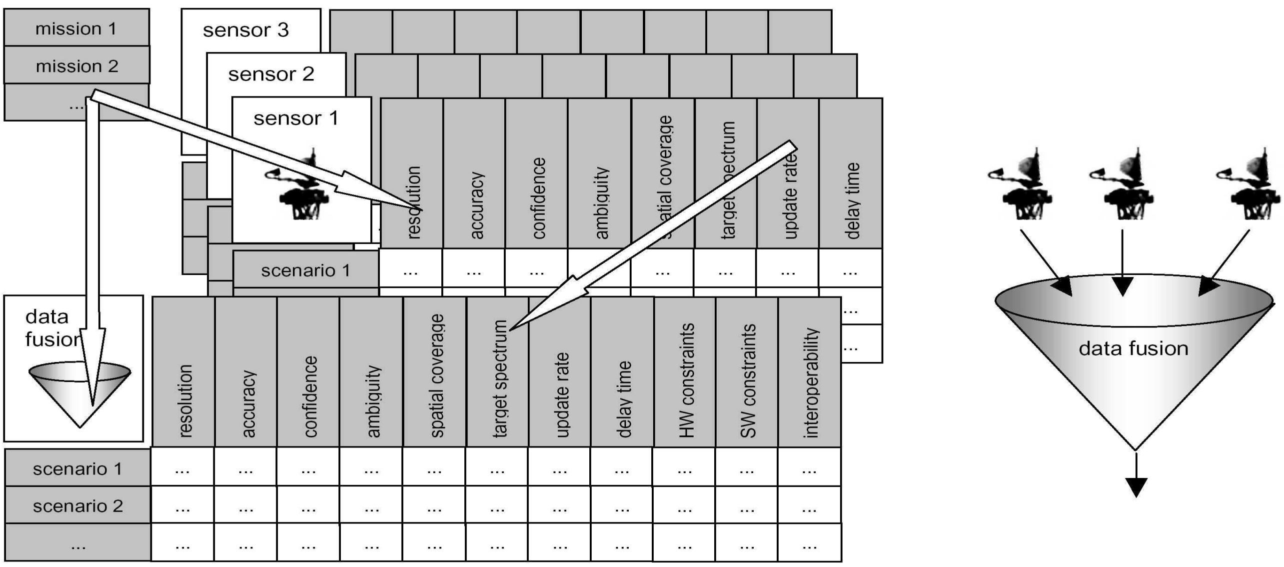


Fig. 3. Data fusion requirements and sensor constraints

In the following we concentrate mainly on level 1 fusion functionality.

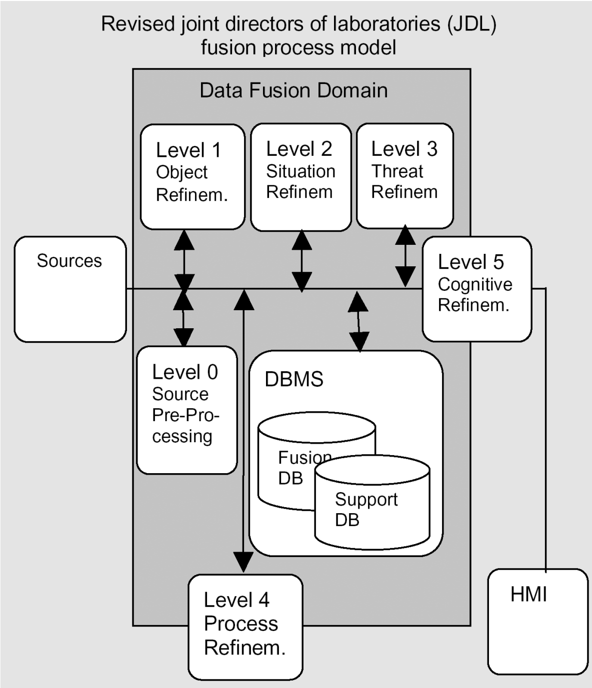


Fig. 4. The JDL fusion process model.

### *2.2.3 Performance requirements*

The necessary performance of data fusion has to be determined. A common approach is to extract the data fusion performance requirements from the flows found in the system analysis/design phase and taking into account the effector requirements. For level 1 data fusion this results in the same points as for the sensor suite: Qualitative and Quantitative Performance. For qualification tests, these requirements are translated into scenarios. These must cover the required target and manoeuvre spectrum (e.g. TBM, missiles, airliners, fighters, helicopters, speedboats, frigates, …) and the sensor target geometry in a realistic way. Figure 5 gives an impression of possible choices. Quantitative requirements are tested by synthetic stress tests. To check the feasibility of the performance requirements special prototyping tools are required. The performance of higher level data fusion aspects is more difficult to measure. In general these aspects need tactical multi-target scenarios.

## 2.3 Software architectural design

The software requirements are translated into software architecture. This is done by a refinement of software items into software components [1].

A data fusion specific approach is to split the fusion process into multiple fusion nodes [10]. Therefore one constructs a (directed) fusion tree, which condenses information more and more until the information suitable for an operator is delivered. The fusion is performed in the different nodes of this tree ("fusion nodes") symbolized by the cone of figure 6. The structure of this tree depends on different interdependent constraints:

- Allocation of hardware
- Communication limitations
- Software controlling
- Data structures
- Distribution on and of platforms

So one has to fulfil hardware and software constraints according to above points: platform autonomy (consistency), communication bandwidth, processor performance, multitasking etc. Further to each node quality and quantitative performance requirements must be allocated, so that the overall data fusion requirements are ensured.

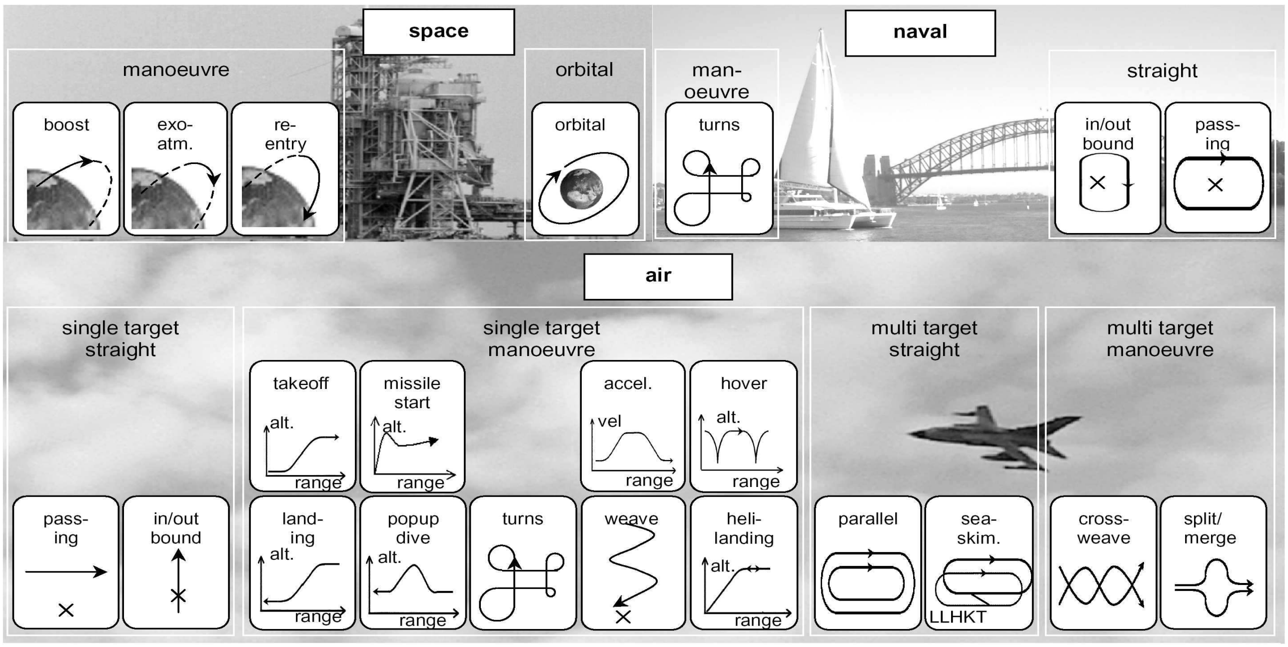


Fig. 5. Target and manoeuvre spectrum.

## 2.4 Software detailed design

In this development phase, a detailed design for each software component is developed. Therefore each software component is split into software units and structured such, that they are completely implementable. Also detailed designs for the interfaces between the software units and data bases are defined. The test of the software units is scheduled and requirements are allocated [1].

Data fusion specific the algorithmic foundations of the fusion nodes are defined. Therefore one distinguishes between level 1 and level 2/3 fusion nodes.

### *2.4.1 Level 1 fusion node*

A level 1 fusion node contains alignment, data association and estimation processing (fig. 6). For data association one differentiates between

- Hard decision methods and
- Probabilistic methods.

Hard decision methods may be nearest neighbour, 2 dimensional data association (e.g. Hungarian, Munkres, Auction, or JVC) or higher dimensional data association (Lagrange Relaxation, Linear Programming). The common realization of a probabilistic approach is the JPD method. For the kinematical estimation one uses today

- Kalman filtering (KF),
- Extended Kalman filtering (EKF),
- Unscented Kalman filtering (UKF)
- Particle Filter (PF)
- IMM techniques.

For attribute fusion (classification and identification estimation) the most popular methods are

- Dempster-Shafer method or
- Rule based estimations
- Expert systems
- Statistical pattern recognition
- Fuzzy logic.

### *2.4.2 Level 2 and 3 node*

The characterisation of a level 2/3 node is more difficult. Up to now there is no practical common accepted approach, how level 2/3 fusion is implemented. In general one has an aggregation which may be viewed as an analogy with the data association in level 1 fusion. Aggregation determines the relations between the different targets (tracks) and estimation corresponds to an assessment of the realized combinations. However, as in the level 1 fusion nodes the information density is also increased. Currently there is no common accepted practical algorithmic foundation in level 2/3. A theoretical mathematical foundation which unifies level 1, 2 and 3 fusion may be founded employing the random set / finite set statistics approach.

## 2.5 Software coding & testing

The software units are translated into source code and the data bases are realized. Afterwards the unit test is performed with special test procedures and data [1].

This step may be significantly enhanced when libraries are available, containing the core algorithms mentioned above, see [6, 11].

## 2.6 Software integration & qualification testing

The software units are integrated into software components and components into the software items according to a integration plan. For each qualification requirement of a software item test cases and procedures are defined and verified in the final qualification testing.

In the language of data fusion, this means that the core algorithms e.g. alignment, data association and estimation are integrated into fusion nodes and fusion nodes are merged into the data fusion system itself . For the test activities a test bed (fig. 7) is used, which realizes a shell around the data fusion kernel including its technical parameters. It covers the topics

- Input data generation and
- Evaluation.

The evaluation topic should contain a HMI with the capability of simultaneous visualization of fusion output against the input data and ground truth.

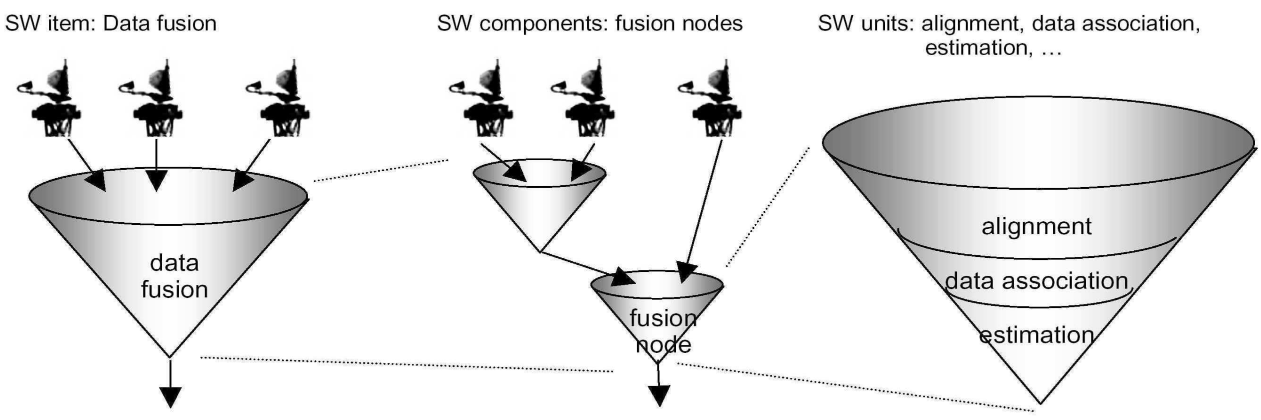


Fig. 6. SW item, components and units for data fusion.

Therefore a first quick analysis is possible to decide about the origin of potential errors. Through the lack of an applicable HMI one spends a lot of time to localize errors in the data fusion part under test where it would be obvious that the true error is produced in the test environment. Using the artificially generated trajectories it is not uncommon that the error lays in the inconsistency of such a trajectory itself. Other errors may be produced by the sensor simulator which does not resemble the real sensor data in an adequate way.

For further evaluation a logging of the fusion result should be produced and evaluated through the test environment. Therefore the evaluation tool should also be capable to provide easy selection of specific data especially for large multi-target cluster situations.

*2.6.1 Input data generation*

The input data generation covers the topics

- scenario generation or GPS recorded trajectories
- sensor simulation

In this first integration step the real sensors are not available. The integration and testing therefore is normally done with scenario generators and sensor simulators. The sensor simulators should give a realistic picture of the real sensors with respect to quality and quantitative constraints. Often one forgets to take the simulation of the delays into account. Many data fusion algorithms are recursive (e.g. filtering). Therefore the behaviour may be totally changed, if the different sensors possess different delay times, i.e. time differences between measurement and data delivery into the fusion process. The simulation of the correct sampling behaviour may be a very complicated task if the sensor suite contains electronically scanned arrays (e.g. active phased array radars). The scenario generator should support the tester to produce realistic trajectories. Therefore one must take into account the possible number and types of targets and their manoeuvrability. Alternatively recorded GPS trajectories could be used to trigger the sensor simulators thus representing target dynamics in a more realistic way.

*2.6.2 The test objectives*

The focus of software integration and qualification test is to provide the succeeding system integration & test phase.

Beside this, it is addressed to the following test items:

- Data Association
- Estimation (filtering, tracking)
- Stress tests
- Fusion of contradicting sensor inputs
- Expensive targets (e.g. TBM and missiles).

Mostly one performs Monte Carlo runs to establish the performance and constraints of the alignment, association and estimation. Monte Carlo runs offer a statistical analysis concerning the estimation and association quality, e.g. stability, loss, and switch of tracks.

Further issues deal with track management (premature deletion, late initialisation/deletion), and tactical picture issues (segmented, redundant, false and missed tracks) [12].

It is important to notice, that not all parts covered by the software integration and qualification test are also contained in the succeeding systems integration and test activity.

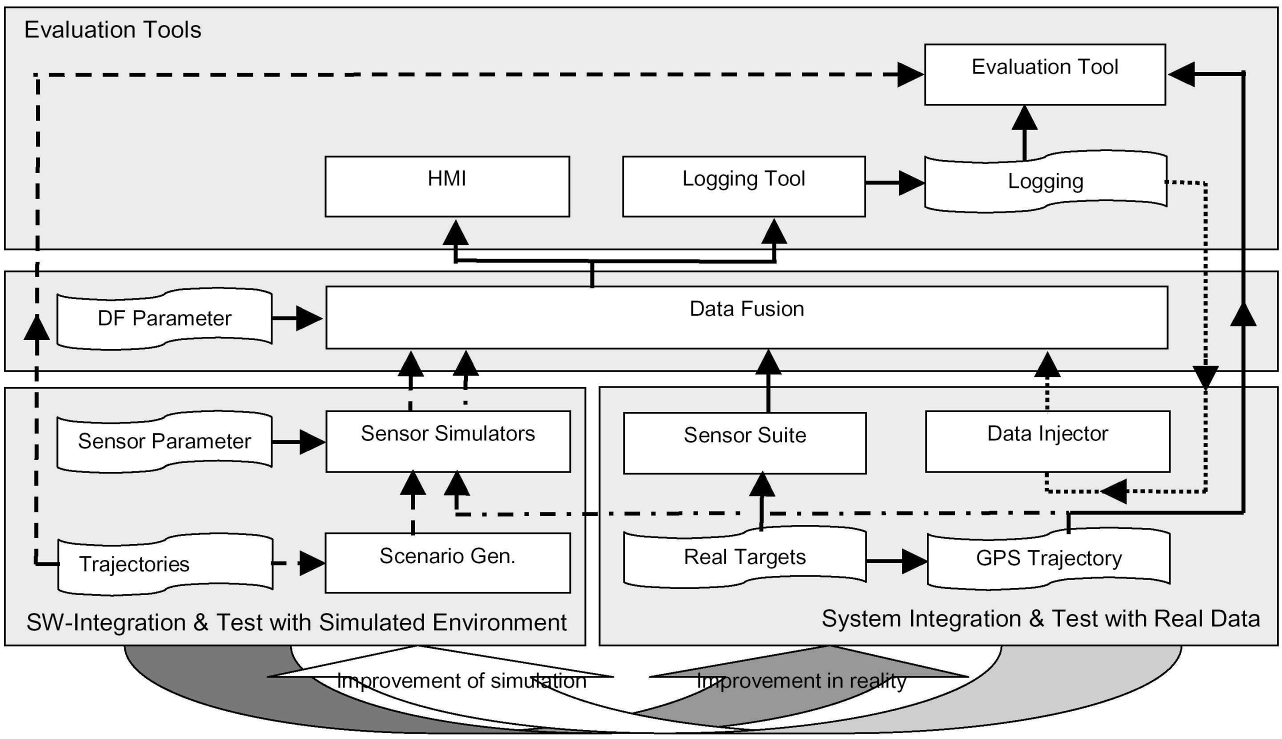


Fig. 7. Software vs. system integration and test.

## 2.7 System Integration & Test

In the system integration and test phase the software items are integrated with hardware items, manual operator interactions and other systems if necessary. Again test cases and procedures are developed and a qualification test is executed [1].

With respect to the development of surveillance systems this means the integration of the data fusion with the sensors and effectors suite and manual operator interactions. This activity takes place on land based test sites, test ranges, on the platforms and may be finalized by tracking tests or even live firing activities for military applications.

### *2.7.1 Input data generation*

In the system integration one uses

- Real sensors and real targets or
- Recordings and data injector tools

instead of sensor simulators and artificially generated trajectories. Recorded loggings are used not only for evaluation but also for tuning of algorithms and technical parameters and regression testing activities. Furthermore not all sensor combinations must be performed with expensive real targets, which helps to reduce the effort. To fulfil this task, it is necessary to have a data injector tool capable to re-inject the recorded sensor data in the system with the same time relations as processed in the live run

### *2.7.2 The test objectives*

The previous software integration & test phase is based on simulation. However, it is out of the scope of a even good simulation to cover all aspects of an data fusion in the reality. The gaps between simulation and reality may occur with respect to the assumption about the

- Sensor capabilities
- Natural environment
- Target attributes and dynamics

### *2.7.3 Sensor capabilities*

The sensor resolution, accuracy, scanning behaviour and internal signal processing resp. tracking may differ from the idealistic assumptions of the previous development phases. So many radar sensors posses a sensor level tracking, e.g. multi hypotheses tracking. This has to be taken into account, especially when track fusion is used by the data fusion system. Modern active phased array radars perform sampling by electronically directing the beam in different directions. This is most times also out of the scope of the simulation capabilities [8, 9, 13, 14].

### *2.7.4 Natural Environment*

A naval sensor platform may be affected by pitch and roll, which must be compensated. The noise in a simulation is generally modelled by Gaussian distributions, which are idealisations of the noise behaviour within the natural environment and real sensors. Measurements may be misaligned by effects which can not completely be considered in the pure simulation approach, like the physical signal propagation. Electromagnetic waves propagating in the atmosphere are refracted or bent. This may extend the radar horizon but influence especially the correct range and elevation measurements. The water vapour evaporated from the sea may cause anomalous propagation effects known as evaporation duct. The impact of this effect depends e.g. on radar wave length and weather condition and is therefore difficult to predict. The detection of targets may be hampered or prohibited by clutter effects [8, 13, 14].

### *2.7.5 Target attribute and dynamics*

There is also a gap between real and simulated target dynamics and manoeuvrability resulting in difficult correlations of its kinematical components.

Information which is closely related to the targets themselves is commonly used for classification and identification and is very important for higher levels of data fusion, e.g. situation and threat refinement.

For radars this information may be linked with high-range-resolution or cross-section fluctuations. The engine modulation caused by turbines or propellers of aircrafts or by hubs and blades of helicopters offers a source of target classification. Other sources may be polarizations or inverse scattering. For cooperative targets the source of identity is secondary radar or IFF, but even for those targets their answers may be corrupted through superposition, whenever they belong to the same segment. Sometimes the answers can not be separated with respect to the targets involved, an effect known as garbling. Targets may also be queried by side lobes, which hampers the precise azimuth measurement. The azimuth measurement may be corrupted by delays in the different transponders. The own IFF interrogator may also receive answers caused by other IFF interrogators, an effect which is called fruit. The altitude information depends on the correct setting of the barometric altimeter in the targets. IFF may be reflected by other targets or objects, so that it is difficult to distinguish between a real and a mirror target. The IFF antenna of the target may be covered by the target itself. Hence missed detections may be experienced depending on the targets aspect angle during movement or manoeuvres. That is the reason, why the future mode S standard requires two antennas [9, 13].

### *2.7.6 Automatic and Human Interactions*

There are manual and automatic interactions, which are normally only testable on the platform. This is e.g. the integration of the data fusion system within the resource management, resulting in automatic sensor (e.g. cuing) and effector (engagement, firing, or warning) activities. On the other side it offers the realistic analysis of manual system maintainability and operator issues. Last but not least the system integration and test issues offer the involved system and software engineers a very valuable chance to collect experiences in the operational handling of the data fusion system.

*2.7.7 Performance Measurement*

The sensor data fusion performance is evaluated for every trajectory and relevant sensor combination against the GPS true path of the target. The improvement against the sensor measurements may be judged via quality factors:

$$Q = \frac{\|\text{GPS positions - fused positons}\|_q}{\|\text{GPS positions - measured positions}\|_q}$$

with $q = 1,2,\ldots,\infty$ and

$$\|(d_1, d_2, \ldots d_n)\|_\infty = \max(d_1, d_2, \ldots d_n)$$

$$\|(d_1, d_2, \ldots d_n)\|_q = \sqrt[q]{\sum_{i=1}^{n} d_i}$$

for a trajectory consisting of n measurements (sample size). A further evaluation is necessary for the course and speed accuracy with respect to the GPS derived data. When direct comparison with measurements is not possible as is normally the case, course and speed values derived by polygonal or polynomial interpolation of the measurement sample are used to judge the filtered course and speed values with respect to the input data (measurements).The association performance is judged upon by evaluation of the result of multi target scenarios (split, merge, parallel, crossing weave and other formations).

## 3 Conclusions

An approach for data fusion development in accordance with different software development standards is given. Especially the test & integration effort for large data fusion systems is discussed by examples gained from practical experiences. It is shown that data fusion development projects need very specific adaptation of the well known project execution standards.


## Acknowledgements

The authors would like to thank the colleagues at EADS, which support of the integration and test activities of the different data fusion projects through the last years. Further we would like to thank A. Haidner and G. Wilmes for their encouraging support, H. Neunert and G. Kouemou for valuable contributions. Last but not least, we would like to thank all the integration & test teams we cooperated with for their hospitality during our stays on the different test sites.


## References


[1] IEEE/EIA. Industry implementation of international standard ISO/IEC 12207, New York, NY, 1998

[2] Department of Defence. DOD-STD-2167 Defense system software development, Washington, D.C., 1988.

[3] Helmut Balzert. *Lehrbuch der Software-Technik: Software-Entwicklung*. Spektrum Akademischer Verlag, Heidelberg, 1996.

[4] Helmut Balzert. *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmenmodellierung*. Spektrum Akademischer Verlag, Heidelberg, 1998.

[5] Elisa Shahbazian, Rodney Hallsworth and Daniel Turgeon. Target tracking and identification issues when using real data, In *Proc. Third Int. Conf. Information Fusion*, Paris, France, 10-13 July 2000.

[6] Werner Henrich, Thomas Kausch, Felix Opitz. Data Fusion for the Finnish Fast Attack Craft Squadron 2000: Concept and Architecture, In *Proc. Seventh Int. Conf. Information Fusion*, Stockholm, Sweden, 28-30 June 2004.

[7] David S. Alberts, John J. Garstka, Richard E. Hayes and David A. Signori. *Understanding information age warfare*. CCRP Publications, 2001.

[8] Albrecht Ludloff. *Praxiswissen Radar und Radarsignalverarbeitung*. Vieweg, Braunschweig/ Wiesbaden, 1998.

[9] S. Blackman and R. Popoli, Modern Tracking System. Artech House, Boston, 1999.

[10] Christopher L. Bowman and Alan N. Steinberg. A systems engineering approach for implementing data fusion system. In David L. Hall and James Llinas editors, *Handbook of multisensor data fusion*, pages 16-1–16-39, CRC Press, Boca Raton, 2001.

[11] Hans Driesen, Frank Mulder, Perry Verveld, and Frank Rulof. MSDF-Lib: A generic Software Library for developing operational target tracking Applications, *NATO RTO Conf. SET-059: Target tracking and sensor data fusion for military observation*, Budapest, Hungary, 15-17 October 2003.

[12] J. Roy and É. Bossé. Definition of performance evaluation methodology for sensor data fusion systems, *DREV-R-9423, Defence research establishment, Valcartier, Québec, Canada, 1995.*

[13] Merill I. Skolnik. *Introduction to radar systems*. McGraw Hill, Singapore, 1981.

[14] Peyton Z. Peebles. *Radar Principles*. Wiley-Interscience Publ., New York, 1998.