A traceable automatic flight state evaluation

Lutz Bretschneider
Institute of Flight Guidance
University of Technology Braunschweig
Braunschweig, Germany
l.bretschneider@tu-braunschweig.de

Peter Hecker
Institute of Flight Guidance
University of Technology Braunschweig
Braunschweig, Germany
p.hecker@tu- braunschweig.de

***Abstract*— Confidence is an important aspect in the context of system automation. To address this topic a lot of research focuses on the design of the right information at the right time and on the integrity of information. Furthermore a lot of work is done in the human machine interface design to depict information right. But only a few work focuses on the way to depict how the information is generated by the systems. One approach is an additional function that translates system functions from the results of the system. Another approach is a system structure that is reliable and traceable itself. This approach is used in the related PhD project dealing with an automatic flight state evaluation. This paper describes the intended system structure with the focus on traceability.**

Flight State Evaluation; System Design; Traceability; Knowledge-Based System; Event-Based Knowledge Activation

I. Introduction

The workload of a flight crew increases (especially in abnormal situations) due to the crowing complexity and the higher level of automation in avionics, the increasing number of actual information and the derived decision making. Even in remote piloted systems the flight state evaluation is an important task for the right decision making process. Motivated by this challenges the PhD project researches in the automation of the flight state evaluation process. This process is currently executed by the human with the support of single task automation (e.g. the engine monitoring, the collision avoidance, etc.). In this context the process is named Situation Awareness. Endsley (1995) defines it as “a state of knowledge, from the processes used to achieve that state”[1]. The model of Situation Awareness consists of 3 parts: a) perception of elements, b) comprehension of current situation and c) perception of future status. All three parts are addressed in the PhD project. To automate this, a complex information analysis is implemented. For the automation of complex tasks confidence of the human in the automation is an important parameter, even in high sensitive areas like the cockpit. The system design influences this in the following ways.

First the given results of the system have to be true in the defined operational limits. If there are no limitations defined the number of true results has to be high as much as possible. This is also addressed in the research of system integrity. Filtering algorithms allow to improve primary information by the use of secondary information. Another aspect in the system design is the time when the results are given. If the human knows that results are available in time at any time the trust is higher than in a system that generates results in unpredictable changing times. Furthermore the system design influences confidence by the way of designing the human machine interface. That allows a presentation of the results in a situation adapted way. This includes coloring, fonts, display structures, menus, display technology, acoustics and interaction methods. Less research is done in the presentation of how the results are collected but especially in complex systems this information is important to trust in the given results. To address this for the automation of complex flight state evaluation the paper describes a system approach that intends a traceable way of collecting results. Because of the focus on anomalies in flight the results will change only a few times but the system traceability is expected to explain the way of how the system works also in non-critical situation.

II. System Architecture

The implementation of the flight state evaluation process is oriented on the definition of Situation Awareness by Endsley (1995). Therefore and with the focus on a traceable system architecture Figure 1 illustrates the intended structure. It shows exemplary the property-graph-based knowledge base, a system ability layer and a layer of interfaces. The figure consists of several simplifications and does not claim on completeness.

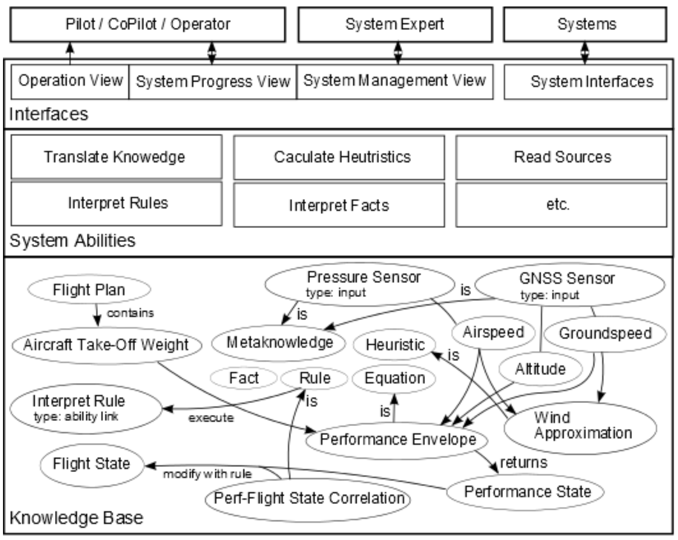

Figure 1 - Intended System Structure

The first important part of the intended system is the knowledge base (II.A) that allows to describe the flight state at any time respectively to describe every situation in flight as accurate as possible. This is the requirement for enabling the situational awareness of the system. The second important part is the knowledge discovery (II.B). Therefore an event-based knowledge activation is intended. It helps to activate the necessary knowledge situation depended. The third important part of the system are the interfaces (II.C). They allow an interaction with the knowledge and its discovery.

A. Knowledge Base

As previously mentioned the knowledge base is the core part of the intended system. It contains procedural and declarative knowledge about the flight and all related information as well as information of the system and its processing. Procedural knowledge is defined as a result of sequences with an expected result and can be compared to human experience given by recurring activities. It is mapped in the form of heuristic knowledge. Declarative knowledge is represented in form of facts and rules [4]. Furthermore both types of knowledge are given by Metaknowledge. It includes predefined heuristics as well as facts like the abilities of the system. Abilities are stored as links in the knowledge base. In a future system an implementation of the function in the knowledge base is possible but not in the scope of the related PhD.

The intended knowledge representation is based on a Property Graph [5]. It combines the idea of semantic networks with the theory of graphs. Data can be represented in a form of nodes, edges respectively relationships and properties (cp. Figure 1). The combination of nodes and relationships results in an information base. The specification of nodes and relationships with properties adds information and results in a definition of knowledge. The characteristic of the Property Graph allows depiction of knowledge with a less defined structure that easily enables a depiction of different kind of entities. Moreover, the principle of a local search, that means search from one node via the relationships to other nodes, implies a scalable performance of knowledge access. The paper "A Property-Graph-Based Knowledge Representation for Decision Support Systems" [3] focuses on the performance. The information-relationship-mapping also propose advantage in the traceability of the system knowledge. This aspect is more detailed in section III.

B. Event-Based Knowledge Activation

With the focus on a situation dependent knowledge activation an event-based knowledge activation is intended. For this task an event is defined as a lockup of one information, that can be a node or relationship in the graph to prove a rule, calculate an approximation or access a fact. The events are triggered by a change or by a definition. For that reason three types of tasks are identified:

a) check content of an information – to identify false data,

b) check related information resulting by a new information state - to identify all consequences from the change of information,

c) check the update of information systems - to identify if systems are running or defined limits are reached.

Task a) and b) are triggered if an information changes. The results of the event will trigger new events until the top level is reached "flight state normal" or the results of an event does not change and not require a further event. The monitoring of the system inputs or predefined limits (cp. task c)) demand predefined trigger of events to detect anomalies without a change. For example if a source does not produce new values and it does not detect the error itself.

To visualize the event-based knowledge activation Figure 1 depicts the airspeed information as a node in the knowledge base. In the case of a changed airspeed the information itself is proven. Therefore the "Wind Approximation" helps to qualify the new airspeed information with the use of the wind velocity vector and the groundspeed. Furthermore the information has influences on the performance envelope. It is evaluated and in the case the resulting performance state does not change no new event is triggered. In the case of a performance state change the event is triggered to interpret the "performance-flight state correlation" rule to update the flight state value. Moreover, the airspeed check also effects the "working state" of the airspeed information that can change the flight state at the end of the event-flow. But this is not illustrated in the figure and is only one simplification. Additionally an error in the airspeed source is detected via an event iteration that is defined to check whether the airspeed information changes or not.

C. System Interfaces

The system interfaces include the interfaces to all data and information systems as well as the interfaces to the human. The interfaces to data sources are assumed as given with a readable access. The interfaces to the human are specified by two user groups.
First the primary user during operation that are pilots, copilot or operators. They are supported by the results of the flight state evaluation process. That includes in abnormal situations alerts, warnings and if necessary background information where the warnings are derived from. Additionally the system enables a progress view that depicts what the system currently focuses on. The second user group are the domain experts (system engineers, maintenance, certification) that are interested in the system performance. It includes the content of the knowledge base as well as event statistics.

III. TRACEABLITY

As mentioned in the introduction the system design is important for the traceability of the automated flight state evaluation process. Therefore the relationship-based information mapping supports the depiction of the dependencies between information in the system. With a method that translates system wordings into human language a readable system is enabled. The relationships with a direction and properties allow a multilayer knowledge management to represent hybrid knowledge in the system. Furthermore the complete modeling of all knowledge in one structure - the property graph - allows a traceable architecture because all information (rules, heuristics, facts and Metaknowledge) can be mapped und used at one point. The fragmentation of knowledge and the lost of information caused by structures or scattered implementation is prevented.

The event-based knowledge activation is expected to enable a situation dependent use of knowledge. With a reliable result the confidence in the system increases. If the results are traceable but not as expected the system improves the situation awareness of the human, e.g. in a high workload situation. To minimize the workload of the human in use of the intended system the concept of a procedure is introduced.

The procedure is defined as an event flow respectively a list of dependent events generated by an definition or a change. As mentioned in section II.B events are triggered by a change or a definition and can result in new events. For example an input of a velocity forces two events and the changed performance state requires further events until the last event sets a new flight state information. Those events are hard to monitor, because they are processed very fast and have only less information. The procedure definition merges those event flows from the input to the last event and slow down the progress information while increasing the information content for the human. Furthermore the procedures can be predefined for checklists that include a number of events that have to be triggered.

In combination with the interfaces the procedure definition can help to minimize the content of the system progress view during operations as well as in the expert view.

IV. Further Work

This paper describes the intended system structure with the focus on traceability. The system is already in the first implementation phase and has to be finalized. During that phase further aspects for a reliable system are identified and investigated. The final validation includes a human in the loop simulation with a questionnaire that focuses on the scoring of increased system traceability during simulated flight operations. The contribution to the increased traceability of the hybrid knowledge, the graph-based knowledge representation, the event-based knowledge activation and the procedure filtered event depiction is analyzed.

References

[1] M. R. Endsley, "Towards a Theory of Situation Awareness in Dynamic Systems", Human Factors, 1995.

[2] L.Bretschneider and P.Hecker, "A knowledge-based flight state evaluation approach using machine learning technology ", ATACCS 2012, London, 2012.

[3] L.Bretschneider and P.Hecker, "A Property-Graph-Based Knowledge Representation for Decision Support Systems ", ATACCS 2013, Naples, 2013

[4] R. A. Akerkar and P.S. Sajja, "Knowledge-Based Systems", Jones and Bartlett Publishers, 2010

[5] What is a Graph Database?, http://www.neo4j.org/learn/graphdatabase, 14.02.2014