



GENERATION AND COMMUNICATION OF STRATEGIC PLANS AT DIFFERENT LEVELS OF ABSTRACTION FOR INTELLIGENT ASSISTANCE SYSTEMS IN GENERAL AVIATION

Prakash Jamakatel¹ & Jane Jean Kiam¹

¹University of the Bundeswehr Munich, Institute of Flight Systems

Abstract

A single-piloted ultralight aircraft implies a lack of safety redundancy. Introducing a pilot assistance system onboard to support and guide the pilot in decision-making, in nominal as well as in non-nominal situations, enables a reduction of the cognitive load, thus increasing flight safety. AI-planning techniques such as Hierarchical Task Network (HTN) planning can be used to generate strategic plans consisting of instructions at multiple levels of hierarchy to guide the pilot. In this paper, we recall how cockpit tasks can be modelled for HTN planners. Subsequently, we describe an approach for the efficient communication of the generated strategic plans as instructions at different levels of hierarchy through visual channels. We evaluate the communication approach using human-in-the-loop tests and find that the proposed approach is viable.

Keywords: AI-planning, Adaptive assistance system, Hierarchical planning, Plan visualisation, Flight guidance, Ultralight

1. Introduction

Ultralight (UL) aircraft have a very limited take-off weight [1], e.g. a maximum takeoff weight (MTOW) of up to 600 kg is allowed in Germany. They can also be piloted by less experienced amateur single pilots, and lack the assistance systems that are ubiquitous in commercial and military aviation. Because of this, UL aircraft are more susceptible to errors induced by human factors. The incorporation of intelligent Pilot Assistance Systems (PASs) in UL cockpits has been identified as an appropriate measure to reduce the number of fatal outcomes resulting from such errors [2]. Such PASs are expected to guide the pilot on how to perform typical tasks in the cockpit, either on request or proactively. Simultaneously, they continuously monitor the pilot's activity in the cockpit so that the guidance provided is contextual, essentially emulating the role of a co-pilot. This makes UL aircraft an ideal test-bed for Single Pilot Operations (SPO).

Based on the assumption that UL-pilots possess different levels of knowledge and, depending on their experience levels, some may not know how to react in in-flight emergency situations resulting from defects of the aircraft components. Furthermore, due to the less strict medical and training requirements, medical conditions or lack of knowledge can cause reduced pilot capabilities, potentially resulting in fatal in-flight accidents. To reduce the risks of fatal accidents, we developed a pilot assistance system called FRICO (i.e. FRiendly COpilot) for single-piloted UL aircraft, where methods from automated planning are applied to assist the pilot, especially in critical emergency situations [3, 4]. Strategic flight plans for the guidance of the pilot to safety are automatically generated using hierarchical planning [5], while automated context determination is performed by inferencing from passively observable data (e.g. tracked gaze, interactions with the cockpit instrument panels and controls) [6], as well as an interactive goal-directed dialog system [7]. Automated context determination is essential to ensure that plans generated for flight guidance are relevant, and are therefore life-saving in emergency situations. However, little has been done to investigate the communication of the generated plans using HTN planners, which are formally in forms of task networks, to the pilot.

This paper introduces an approach to communicate the assistance generated using HTN planners in the form of strategic plans to the pilot. We first provide an overview of the related work in Section 2. This is followed by a brief background on our previous works on the system architecture of FRICO and how HTN planning can be used to model typical cockpit tasks used as domain knowledge to generate strategic plans in Section 3. Subsequently, we describe our recent work on communicating the strategic plans as instructions at different abstraction levels in Section 4. A preliminary evaluation of the proposed approach is provided in Section 5 to obtain feedback from users on their acceptance of the developed assistance system.

2. Related Work

A single-pilot cockpit with onboard support is seen as a possible means for a reduction of operation and labour costs while maintaining the utility of the dual-pilot cockpit in commercial aviation [8]. Several modalities for SPOs exist, as shown in a survey in [9]. One popular modality is to replace the second pilot with automation and ensure that automation is capable of taking over the tasks of the second pilot and that the omission of the second pilot does not induce a higher workload on the remaining pilot. Examples include a Cognitive Pilot-Aircraft Interface (CPAI) presented in [10], and a support system introduced in [11]. In [10] cognitive state estimation is combined with physiological sensing to facilitate pilot-aircraft interaction and prevent human pilot errors.

Since assistance systems in aviation aid the pilot in performing various cognitively demanding tasks like planning and decision-making in an automated fashion, the trend has been towards incorporating advanced AI techniques into the assistance systems to augment their capability. Approaches based on knowledge-based systems [12], fuzzy logic and expert systems [13], Hidden Markov Models (HMMs) [14] and probability theory in combination with automated planning [4] can be found. In the related subdomain of automated driver assistance systems, various machine learning approaches have been proposed [15, 16].

In [17], an approach to provide timely assistance and relevant flight information to the pilot according to the flight context was studied. This approach leverages speech-based interaction, where the pilot can either request the information directly or is provided information via contextual notifications. The abstraction level of the provided information is not discussed in the paper. However, it is desirable that the abstraction level of the generated assistance should be adaptive, depending on the experience level and current mental workload of the pilot. Automated hierarchical planners are well-suited to generate plans that fulfil the aforementioned criteria and the investigation of their usability in the aviation domain has recently gained traction.

According to Ghallab et al., automated planning is a branch of artificial intelligence that deals with the process of “choosing and organizing the actions that can achieve the given objective” [18]. An *action*, in automated planning, is something done deliberately by an agent, i.e. any entity capable of interacting with its environment, in order to change its own state or the state of its environment [18]. The domain knowledge is encoded into automated planning problems using domain description language such as PDDL [19] and can be solved using domain-independent planners such as the Fast Downward planner [20]. Classical planning is the most popular flavour of automated planning where finite, static and deterministic world-assumptions are made without the notation of time [18]. Hierarchical Task Network (HTN) planning is an extension of the classical planning problem, where hierarchical structures can be included in the modelling of the problem domain. In UL cockpits, where one form of assistance to be provided consists of the synthesis of a sequence of actions as a plan suggestion that the pilot can carry out to reach his or her current flight objective or intent, automated planning lends itself to be a suitable method. By using automated planning, pre-existing domain knowledge from Pilot Operating Handbooks (POH) [21, 22] can be efficiently integrated into assistance generation.

An execution assistance system called CHAP-E based on hierarchical planning that guides the pilot during various flight phases on the best course of action according to the state of the environment has been introduced in [23]. In [24], plan and goal recognition as planning has been used to first infer the pilot’s intent and to assist the pilot in performing the intent subsequently. In other domains, hierarchical planning has been used to generate instructions at multiple levels of hierarchy in [25] and for the generation of instructions for Do-It-Yourself in home improvement setting [26].

Various plan visualisation approaches have been developed in the literature. A web tool to visualize planning domain and solution plans has been introduced in [27]. An interactive visualization tool for plans based on the Gantt chart has been introduced in [28]. A tool for the communication of the branching plans (plans with multiple branches that can lead to the goal), where the plan is communicated as a radial graph for human-agent decision-making, has been developed in [29]. Similarly, a tool to visualize HTN has also been developed¹. These desktop tools, however, are not designed for the interactive communication of plans at different abstraction levels in an UL cockpit and are thus not applicable in our use case. The interactive communication allows the pilot to self-determine at which level of detail the strategic plans are communicated.

3. Background

3.1 FRICO

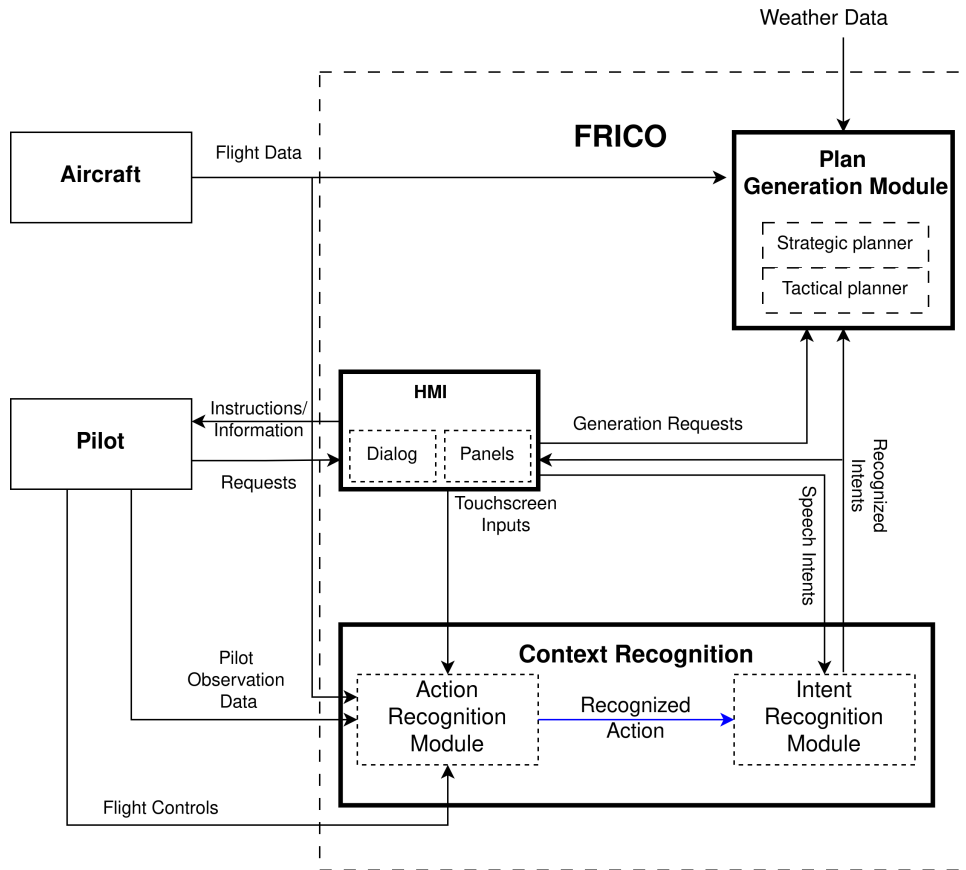


Figure 1 – System Architecture of FRICO [4]

FRICO, an AI-enabled PAS, has been developed in [3]; its ability to recognise the actions performed by the pilot and to infer the pilot's intent for performing these actions has been demonstrated in [6]. FRICO consists of functional modules for context recognition, plan generation and human-machine interfacing (HMI) (Figure 1). Intent recognition in the form of goal tasks and action recognition are submodules that constitute the Context Recognition Module (CRM). The CRM is also complemented by the HMI for acquiring information on the current flight context from the pilot via a cockpit dialog [7]. In contrast, the Plan Generation Module (PGM) comprises modules for computing strategic and tactical plans [4]. Action recognition and tactical plan generation are carried out using a Bayesian state estimation method and numeric planning respectively; strategic plan generation is performed using Hierarchical Task Network (HTN) planning [5], and intent recognition is carried out using techniques from plan and goal recognition as hierarchical planning [30], while basing on the same HTN model used for strategic planning. The HMI consists of dialogues and panels (glass cockpit and informa-

¹https://github.com/Maumagnaguagno/HTN_Plan_Viewer/

tion panels), and is used to provide the pilot with flight-related information. The pilot's flight-related requests are also received through the HMI before being processed.

3.2 Hierarchical Task Network Planning for Strategic Plan Generation

We consider the problem of generating a strategic plan to guide the pilot on performing typical cockpit tasks according to the POH. The substeps to execute the strategic plan that are related to each other or are needed to be performed during a particular flight stage can be logically grouped together. Depending on the experience level of the pilot, either each individual substep within the strategic plan is required to instruct the pilot or a higher level step is adequate. Furthermore, there can be constraints on the ordering in which individual substeps should be executed. This necessitates the generation of plans at multiple levels of hierarchy with ordering information and can be accomplished by using the Hierarchical Task Network (HTN) planning. HTN planning is an extension of classical planning. Unlike classical planning, in HTN planning, the planner's objective is described not in terms of a goal-state that has to be achieved, but instead as a collection of tasks to be performed. Starting from an initial task network, decomposition is carried out recursively until the primitive tasks have been reached. We now explain the modelling of cockpit tasks and HTN terminology with the help of a simplified model of the task "land airplane" (see Figure 2).

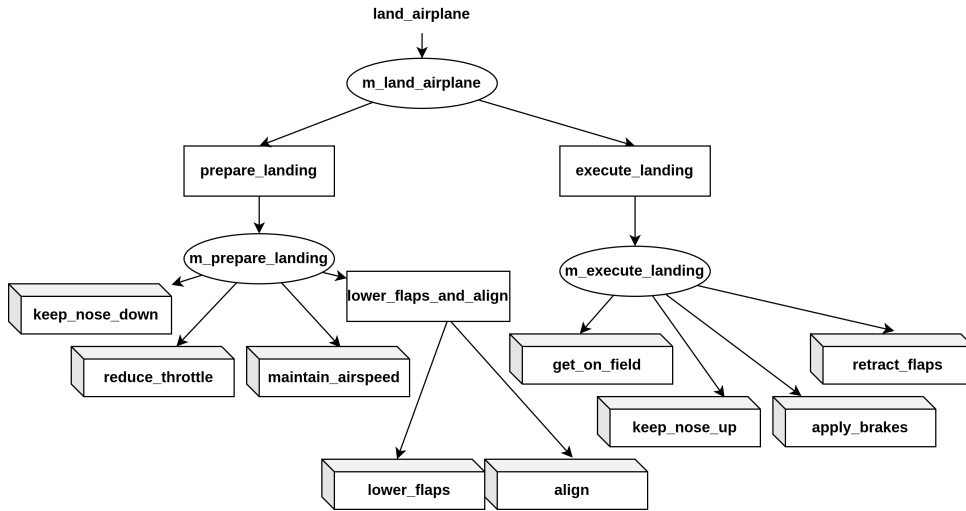


Figure 2 – Example procedure to perform the task "land airplane"

"Land airplane" is the identifier of the task to be decomposed into two compound subtasks: preparing the landing and executing the landing. In Figure 2, the tasks inside a rectangle are the so-called compound tasks. These can be further decomposed into lower-level compound tasks or into primitive tasks (encompassed in cubes in the figure) using a corresponding decomposition method² (represented by oval circles in Figure 2). Primitive tasks are tasks that can be executed by the agent. Solving the planning problem entails finding out the complete decomposition structure given an initial task identifier, such that the leaf nodes upon final decomposition consist only of primitive tasks. The planner performs this by selecting the appropriate decomposition method for each compound task within the task network.

Figure 3 depicts a snippet of the encoding of the example procedure from Figure 2 using the Hierarchical Domain Description Language (HDDL) [31]. By using this syntax, we can describe the ordering information between the tasks, preconditions, and decompositions. We refer to [5] for detailed information on the HDDL model of the UL domain.

4. Communication of Generated Plan

Once the generation of the strategic plan is complete, the generated plan has to be communicated to the pilot. Typically, the plan can be generated using the progression search algorithm. The algorithm

²A compound task can have multiple decomposition methods. This allows the encoding of information that several strategic plans be possible for one task.

```

(:method m_land_airplane
:parameters()
:task(land_airplane)
:precondition(and())
:subtasks
(and(task1(prepare_landing))
(task2(execute_landing)))
:ordering(and(task1 < task2)))

(:method m_prepare_landing
:parameters ( ?reduceThrottle - ReduceThrottle)
:task(prepare_landing)
:precondition (and )
:subtasks (and
(task1(keep_nose_down))
(task2(reduce_throttle ?reduceThrottle))
(task3(maintain_airspeed ?airspeed))
(task4(lower_flaps_and_align ?flapState ?alignState))))

(:method m_execute_landing
:parameters ( )
:task(execute_landing)
:precondition (and (p_isPerformed ?prepareLanding))
:ordered-subtasks
(and(task1(get_on_field ?location))
(task2(keep_nose_up ?noseState))
(task3(apply_brakes ?brakeSpeed))
(task4(stop_airplane ?airspeed))))
    
```

Figure 3 – A snippet of the HDDL encoding of the task network presented in Figure 2

incrementally decomposes and executes the task, starting from the initial state towards the goal. The state is continuously updated to reflect the effect of the execution. The progression search algorithm within the PANDA framework³ can be parameterized such that either the plan with the lowest total cost (each primitive task is associated with a cost value), or lowest decomposition length is generated. We refer to [32] for a detailed description of the algorithm. The output of the progression search contains a list of primitive tasks along with their identifiers and arguments, the root task and the decomposition methods applied in order to obtain the primitive tasks. Thus, by controlling which decomposition is shown, the level of hierarchy can be controlled. However, using solely the output of PANDA, it is unclear whether the applied decomposition method is totally ordered or partially ordered⁴. Therefore, it is not possible to infer the ordering of the tasks (to be communicated as instructions to the pilot). Since the ordering information is essential for plan communication, to extract the ordering information, we transform the generated plan using the progression search algorithm into a plan verification problem within the PANDA framework. Then using the SAT solver, the output of which contains the ordering information between the primitive tasks extracted from the input hierarchical task model (defined in HDDL), relative orderings of the tasks can be extracted. The SAT solver finds a solution to the planning problem by transforming the planning problem into a boolean satisfiability problem. Refer to [33] for detailed information on how an HTN planning problem can be solved using a SAT-based approach.

4.1 Plan Visualisation

With the ordering information and instructions (i.e. subtasks which are either compound or primitive tasks) about the solution plan, inspired by Gantt chart-based visualisation [28], we develop a visualisation system with colour-coded display elements on decomposability, as shown in Figure 4. Atomic steps (or primitive tasks in the solution plan) within the strategic plans are shown inside a green rectangle, while decomposable steps (i.e. compound task) are shown inside a purple rectangle that has a white triangle in the lower right corner (see AN_2 in Figure 4a). The horizontal axis assimilates the timeline, i.e. we display currently executable actions on the left side of a scrollable strategic plan

³<https://github.com/panda-planner-dev>

⁴Refer to the IPC output format documentation for detailed information: <http://ipc2020.hierarchical-task.net/data/format.pdf>

display unit on our interface (see AN_1 in Figure 4a). The plan display unit can be scrolled either vertically or horizontally to fit the content in the view. The user can zoom out using a pinching gesture to fit the entirety of the plan into the plan display unit. If multiple subtasks are currently applicable (i.e. they are partially ordered), then they are stacked vertically. The information on total ordering (i.e. which subtask has to be performed before which subtask) is represented using a linear dependency visualisation, where the subtask at the start of the line has to be performed before the subtask at the end of the line.

If necessary, the pilot can click on the subtask denoted in blue to expand the plan with details on how to perform the particular subtask. The expanded view is enclosed within a rectangle and is connected to the parent subtask (see 4b for an expanded view of subtask AN_2) at a lower hierarchy level. The expanded view can be collapsed by clicking on the higher-level subtask again. If more than one subtask is expanded at any given time, only the hierarchy of the most recently clicked subtasks is highlighted, while reducing the opacity of the previously expanded subtasks. This design decision is taken so that the HMI is not overfilled with information and based on the assumption that the pilot is mostly interested in the detailed decomposition of the subtask on which he/she currently clicks.

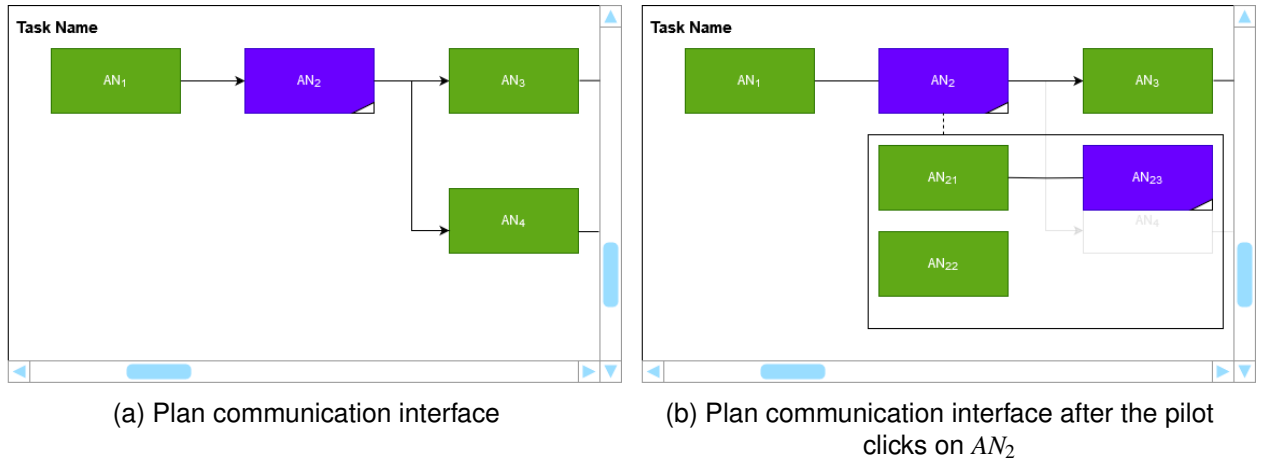


Figure 4 – Design of the plan communication interface

5. Preliminary Evaluation

The preliminary evaluation of the proposed approach for the generation and communication of the strategic plans in different layers of hierarchy is performed in a realistic UL flight simulator. We choose four different flight missions for evaluation, two nominal (i.e. landing the aircraft and taking off) and two non-nominal (i.e. reacting to an engine fire and reacting to an electric fire).

5.1 Simulator



(a) Simulator



(b) HMI with generated strategic plan displayed

Figure 5 – Simulator and HMI

Question	Question label
The generated hierarchy makes sense to me.	Q1
It is clear to me, how to switch between different hierarchy levels.	Q2
The ordering information between the subtasks is clear to me.	Q3
The user interface is intuitive to use.	Q4
The provided visualization of the strategic plan was clearly understandable.	Q5
The control gestures are intuitive.	Q6
The generated strategic plan helped me during the execution of the task.	Q7

Table 1 – Questions included in the Likert scale questionnaire

Figure 5a demonstrates the simulator used for our evaluation. We use the commercially available flight simulator X-Plane 11⁵ (i.e. Aerolite 103 aircraft model) to simulate the flight mission. The flight data from X-Plane is communicated to the other modules using the ROS backbone defined in [4]. The outside view is projected using a 75-inch external display, while the glass cockpit and interaction interface are displayed on a touchscreen. We use Thrustmaster Warthog Hotas⁶ as flight stick and throttle. The simulator skeleton is built using the Next Level Racing GTTRACK⁷.

5.2 Experimental Design

To determine the usability of the designed interface, we conduct a preliminary evaluation. Five participants (4 males and 1 female) are prompted to execute the aforementioned flight missions. The flight missions are triggered in X-Plane using the so-called situation files. The situation files capture all the necessary details including aircraft position and orientation as well as aircraft configuration among others⁸. The test persons are briefed on the workflow of the system. Subsequently, the test person carries out the four flight missions while the strategic plan on how to perform the flight mission is generated and communicated to the test person. Initially, instruction is provided without substeps and can be expanded by the test person if desired. Upon completion of the mission, the test person is instructed to fill out the 5-point Likert scale evaluation form. Specifically, we ask questions about the generated strategic plan, the user interface and the overall perceived helpfulness of the system.

5.3 Results

Table 1 lists the questions we used in the Likert scale questionnaire. The result of the evaluation is depicted in Figure 6. As seen in the Figure, the designed interface is perceived as intuitive by all test subjects and all test subjects at least agree that the generated instruction helped them during the execution of the task with a mean score of 4.4. The clarity of the ordering information was reported to have a mean score of 4.0. The instruction generated by the planner for each of the cases was within 10 ms for each test mission and subsequent decompositions.

6. Conclusion and Future Work

This work focuses on the communication of strategic plans generated using an HTN planner as instructions at multiple abstraction levels. These plans provide flight guidance on how to perform typical cockpit tasks. The communication of the generated instruction is carried out through visual channels. The designed system can represent the mutual ordering of the instructions and their subsequent dependent instructions⁹ within the strategic plan.

In future works, various communication channels for the communication of the generated strategic plan should be explored. For example, if the mental state of the pilot allows them to receive the

⁵<https://www.x-plane.com/>

⁶<https://www.thrustmaster.com/products/hotas-warthog/>

⁷<https://nextlevelracing.com/products/next-level-racing-gttrack/>

⁸<https://x-plane.helpscoutdocs.com/article/66-x-plane-11-10-situation-file-improvements>

⁹These instructions should only be performed once the previous instructions have been completed and the additive effects of the previous instructions satisfy the preconditions of the current instruction to be performed. A single instruction is equivalent to the corresponding action of a strategic plan.

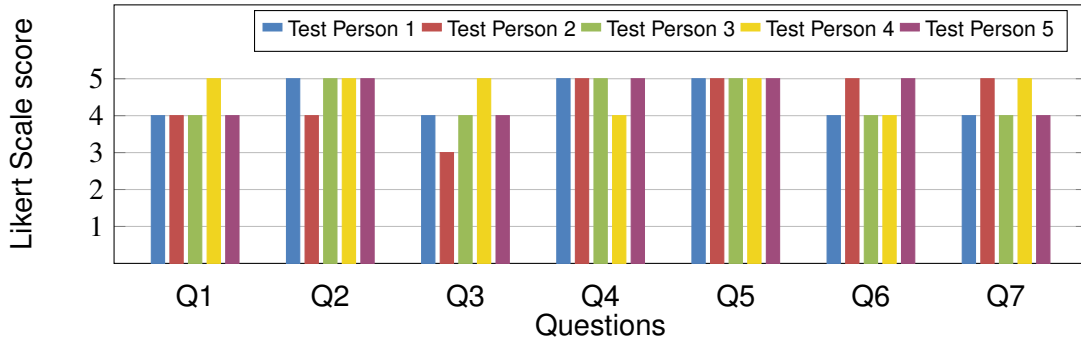


Figure 6 – Result of the evaluation on Likert scale 1-5 (1; Completely Disagree, 5; Completely Agree), n = 5

instruction more efficiently through an auditive channel, then this channel can be prioritized to instruct the pilot on instantaneous actions. Similarly, instead of mapping the name of the tasks (as modelled in HDDL) of a strategic plan to the displayed instructions directly, natural language mapping can be explored to make the instructions more intuitive. Similarly, human-in-the-loop tests can be extended to include more test subjects. This will help to better assess the system's usability.

Acknowledgement

We acknowledge the help provided by Dr. Gregor Behnke during the application of PANDA framework. This work is funded by the German Federal Ministry of Economic Affairs and Climate Action (Project MOREALIS).

Contact Author Email Address

prakash.jamakatel@unibw.de
jane.kiam@unibw.de

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

- [1] Council of European Union. Council regulation (EU) no 748/2012, 2012.
- [2] BFU. Studie zur Flugsicherheit von Luftsportgeräten - Analyse von Unfällen und Störungen mit Luftsportgeräten in Deutschland in den Jahren 2000-2019. Technical Report BFU22-803.1, Bundesstelle für Flugunfalluntersuchung, 2022.
- [3] Prakash Jamakatel, Sondes Morchedi, and Jane Jean Kiam. FRICO: An AI-enabled friendly cockpit assistance system. In *ICAPS 2023 System Demonstrations*, July 2023.
- [4] Prakash Jamakatel and Jane Jean Kiam. A system level overview of FRICO – a single-pilot cockpit assistance system. In *IEEE International Conference on Human-Machine Systems (ICHMS)*, May 2024. IEEE International Conference on Human-Machine Systems (ICHMS).
- [5] Jane Jean Kiam and Prakash Jamakatel. Can HTN planning make flying alone safer? In *Proceedings of the 6th ICAPS Workshop on Hierarchical Planning (HPlan 2023)*, pages 44–48, 2023.
- [6] Prakash Jamakatel, Pascal Bercher, Axel Schulte, and Jane Jean Kiam. Towards intelligent companion systems in general aviation using hierarchical plan and goal recognition. In *International Conference on Human-Agent Interaction, HAI 2023, Gothenburg, Sweden, December 4-7, 2023*, pages 229–237. ACM, 2023.

- [7] Prakash Jamakatel, Rebecca de Venezia, Christian Muise, and Jane Jean Kiam. A goal-directed dialogue system for assistance in safety-critical application. In *International Joint Conference on Artificial Intelligence*, 2024.
- [8] Jonathan Graham, Chris Hopkins, Andrew Loeber, and Soham Trivedi. Design of a single pilot cockpit for airline operations. In *2014 Systems and Information Engineering Design Symposium (SIEDS)*, pages 210–215. IEEE, 2014.
- [9] Guoqing WANG, Min LI, Miao WANG, and Dongjin DING. A systematic literature review of human-centered design approach in single pilot operations. *Chinese Journal of Aeronautics*, 36(11):1–23, 2023.
- [10] Jing Liu, Alessandro Gardi, Subramanian Ramasamy, Yixiang Lim, and Roberto Sabatini. Cognitive pilot-aircraft interface for single-pilot operations. *Knowledge-based systems*, 112:37–53, 2016.
- [11] Paul Schutte, Kenneth Goodrich, and Ralph Williams. Synergistic allocation of flight expertise on the flight deck (SAFEdeck): A design concept to combat mode confusion, complacency, and skill loss in the flight deck. In *Advances in Human Aspects of Transportation: Proceedings of the AHFE 2016 International Conference on Human Factors in Transportation, July 27-31, 2016, Walt Disney World®, Florida, USA*, pages 899–911. Springer, 2017.
- [12] Michael Strohal and Reiner Onken. Intent and error recognition as part of a knowledge-based cockpit assistant. In Steven K. Rogers, David B. Fogel, James C. Bezdek, and Bruno Bosacchi, editors, *Applications and Science of Computational Intelligence*, volume 3390, pages 287 – 299. International Society for Optics and Photonics, SPIE, 1998.
- [13] Jie Rong, Theresa Spaeth, and John Valasek. Small aircraft pilot assistant: Onboard decision support system for SATS aircraft. In *AIAA 5th ATIO and 16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, page 7382, 2005.
- [14] Stefan Suck and Florian Fortmann. Aircraft Pilot Intention Recognition for Advanced Cockpit Assistance Systems. In *Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience*, pages 231–240. Springer, Cham, Switzerland, June 2016.
- [15] Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, and Fei-Yue Wang. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE transactions on Vehicular Technology*, 68(6):5379–5390, 2019.
- [16] Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhausen. TransDARC: Transformer-based driver activity recognition with latent space feature calibration. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 278–285. IEEE, 2022.
- [17] S. Estes, Kevin J. Burns, J. Helleberg, Kevin Long, Jeffrey Stein, and M. Pollack. Digital copilot: Cognitive assistance for pilots. *AAAI Fall Symposia*, 2016.
- [18] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016.
- [19] Maria Fox and Derek Long. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20:61–124, 2003.
- [20] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.
- [21] Dorothy Pooley. *POOLEYS Private Pilots Manual: JAR Flying Training, Volume 1*. POOLEYS, Cranfield, UK, 2003.
- [22] SHARK. *Flight Manual: UL airplane*. SHARK.AERO CZ s.r.o., 2017.
- [23] J Benton, David Smith, John Kaneshige, Leslie Keely, and Thomas Stucky. CHAP-E: A plan execution assistant for pilots. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.
- [24] Prakash Jamakatel, Pascal Bercher, Axel Schulte, and Jane Jean Kiam. Towards intelligent companion systems in general aviation using hierarchical plan and goal recognition. In *Proceedings of the 11th International Conference on Human-Agent Interaction (HAI 2023)*. Association for Computing Machinery, 2023.
- [25] Arne Köhn, Julia Wichlacz, Álvaro Torralba, Daniel Höller, Jörg Hoffmann, and Alexander Koller. Generating instructions at different levels of abstraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2802–2813, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.
- [26] Pascal Bercher, Gregor Behnke, Matthias Kraus, Marvin Schiller, Dietrich Manstetten, Michael Dambier, Michael Dorna, Wolfgang Minker, Birte Glimm, and Susanne Biundo. Do it yourself, but not alone: companion-technology for home improvement—bringing a planning-based interactive diy assistant to life. *KI-Künstliche Intelligenz*, 35(3-4):367–375, 2021.
- [27] Maurício C Magnaguagno, Ramon Fraga Pereira, Martin D Móre, and Felipe Meneguzzi. Web planner:

- A tool to develop, visualize, and test classical planning domains. In *Knowledge Engineering Tools and Techniques for AI Planning*, pages 209–227. Springer, 2020.
- [28] Roman Barták. Interactive visualization in planning and scheduling. *Knowledge Engineering Tools and Techniques for AI Planning*, pages 157–172, 2020.
 - [29] Julie Porteous, Alan Lindsay, and Fred Charles. Communicating branching plans for human-agent decision making. In *ICAPS 2021 Workshop on Explainable AI Planning*, 2021.
 - [30] Daniel Höller, Gregor Behnke, Pascal Bercher, and Susanne Biundo. Plan and goal recognition as HTN planning. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 466–473. IEEE, 2018.
 - [31] Daniel Höller, Gregor Behnke, Pascal Bercher, Susanne Biundo, Humbert Fiorino, Damien Pellier, and Ron Alford. HDDL: An extension to PDDL for expressing hierarchical planning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9883–9891, 2020.
 - [32] Daniel Höller, Pascal Bercher, Gregor Behnke, and Susanne Biundo. HTN planning as heuristic progression search. *Journal of Artificial Intelligence Research*, 67:835–880, 2020.
 - [33] Gregor Behnke, Daniel Höller, and Susanne Biundo. Finding optimal solutions in HTN planning—a SAT-based approach. In *IJCAI*, pages 5500–5508, 2019.