Use-Case Informed Task Analysis for Secure and Usable Design Solutions in Rail

Amna Altaf[1], Shamal Faily[2], Huseyin Dogan[1], Alexios Mylonas[3], and Eylem Thron[4]

[1] Bournemouth University, Poole, UK
{aaltaf,hdogan}@bournemouth.ac.uk
[2] Robert Gordon University, Aberdeen, UK
s.faily@rgu.ac.uk
[3] University of Hertfordshire, Hatfield, UK
a.mylonas@herts.ac.uk
[4] CCD Design & Ergonomics Ltd, London, UK
eylem.thron@designbyccd.com

Abstract. Meeting secure and usable design goals needs the combined effort of safety, security and human factors experts. Human factors experts rely on a combination of cognitive and hierarchical task analysis techniques to support their work. We present an approach where use-case specifications are used to support task analysis, and human failure levels help identify design challenges leading to errors or mistakes. We illustrate this approach by prototyping the role of the European Railway Traffic Management System (ERTMS) - Signaller, which provides human factors experts a chance to work in collaboration with safety and security design experts.

Keywords: Use-Case; Task Analysis; Cognitive Task Analysis; Hierarchical Task Analysis; Human Factors; Security-by-Design.

1 Introduction

Traditionally, the rail infrastructure is built around safety and human factors. However, as the rail information infrastructure becomes integrated with operational technology, especially with the implementation of European Railway Traffic Management System (ERTMS), new vulnerabilities are introduced leading to new threats that exploit them. As such attacks are directly or indirectly responsible for compromising safety, cyber security as well has become a new concern for rail safety engineers. This emphasises the growing need for achievement of usable security and safety for system efficiency within critical infrastructure of rail [19].

Security risks might originate from hidden vulnerabilities within design of system [5]. Integrated tools would help experts by considering and visualising security risks, safety hazards, and human failures - particularly as security mishaps can result from the latter [27]. These human failures stem from errors, mistakes or lapses which are also the determining factors for human performance [23]. As such safety engineers, focus on identifying all potential hazards as a result of security risks and human failures [5].

The open-source Computer Aided Integration of Requirements and Information Security (CAIRIS) platform has previously been used in conjunction with Human Factors Analysis and Classification System (HFACS) to determine safety hazards and human factors issues [3]. This tool-support aids efficient and systematic analysis, leading to better design decisions. Human factors experts typically rely on Task Analysis (TA) as one of the many approaches for making design decisions based on human performance [1]. The application of appropriate TA tool to ensure automation and efficiency is crucial for design analysis that accounts for human factors.

In this paper, we present an approach where User Experience (UX) techniques are used to conduct TA with CAIRIS, using a combination of Cognitive Task Analysis (CTA) and Hierarchical Task Analysis (HTA). CTA identifies different types and values of cognitive reactions, which influence human performance during completion of tasks. HTA helps identify task dependencies and sequences as a hierarchy, where high-level use cases are refined into low-level use cases. Using the use-case specifications format, relevant cognitive reactions i.e., vigilance, situation awareness, workload, stress, and risk awareness, are scored and used to visualise HTA models. Different levels of human failures are then identified. By using use case exceptions and the HFACS framework, the use cases with highest level of human failures can be categorised and used to identify associated safety and security design solutions in the form of risk analysis. To demonstrate our approach, we have prototyped the role of Signaller using ERTMS.

The rest of the paper is structured as follows. Section 2 describes the related work and Section 3 describes our approach. Our approach is demonstrated by using ERTMS specifications in Section 4. This is followed by discussion and conclusion for future directions of our work in Sections 5 and 6.

2 Related Work

2.1 Task Analysis Processes and Tools

Tasks are performed by users to achieve goals. These are assumptions made about the behavioural specifications of users involved and how they are supposed to interact with the system [14]. Task Analysis (TA) is used to determine the set of tasks to be performed by users under observation. The TA is conducted by identifying the task for analysis, determining the associated sub-tasks and writing a step-by-step narrative for sequence of actions to be performed [1].

There are two main types of TA: hierarchical and cognitive task analysis [14]. The Hierarchical Task Analysis (HTA) is conducted to determine the hierarchy of tasks by decomposing high-level into low-level tasks [11]. The Cognitive Task Analysis (CTA) focuses on the cognitive load put by tasks on users depending on their cognitive abilities [25]. The most notable techniques used for eliciting data for TA are: interviews, focus group discussions, surveys, workshops and questionnaires.

The decisions about design, training needs, human error analysis, stress and workload management are dependent on TA [15]. The human factors experts aim to identify human error sources for resolving human factors issues. As these human error sources are considered determining factors for risk and safety analysis during accident investigations [16]. A Training Needs Analysis (TNA) and mental workload behind tasks can also be used to identify the training gaps to train operators interacting with a system.

The TA approaches are used by human factors experts to identify the system design and engineering requirements. Software tools to support TA include Human Factors Workbench (HFW), Predictive Human Error Analysis (PHEA), and Performance Influencing Factors Analysis [16]. The Human Factors Risk Manager (HFRM) also supports risk scoring, failure modes, and the capture of error descriptions [16].

Table 1. Methods and Tools for Task Analysis with Applications

Task Analysis Method	***Tool-Support***	***Application***
Hierarchical Task Analysis (HTA) [16]	Human Factors Risk Manager (HFRM), Human Factors Workbench (HFW)	Risk Scoring, Failure Mode, Error Description
Cognitive Task Analysis (CTA) [29]	Applied Cognitive Task Analysis (ACTA)	Cognitive Demand & Skill, Training Recommendation, Interface Improvement
Ecological Task Analysis (ETA) [13]	-	Control Theory, Cognitive Psychology
Operator Action Event Tree (OAET) [15]	Event Tree (Success & Failure)	Human Reliability Assessment
Flow Diagram [15]	Flow Chart	Binary Decision Logic
Influence Modelling and Assessment System (IMAS) [15]	Cause-Consequence Model	Skills Diagnostic, Mental Model
Critical Action and Decision Evaluation Technique (CADET) [15]	Critical Action or Decision (CAD)	Potential Cognitive Error, Failure Scenario

Computer Aided Software Engineering (CASE) tools and components are typically used for modelling, e.g. the Unified Modelling Language (UML), scenario-based design and Concur Task Trees (CTT) [14]. The UML pre-defined specification formats in the form of use cases are used to describe actor/s, specific conditions, steps and exceptions for TA, but is limited to data representation. CTT helps in the comprehension of hierarchical task breakdown, representation of activities using graphical syntax, and task allocation including attributes, but it lacks in understanding the cognitive attributes (i.e. mental workload) needed to complete tasks.

A brief summary of TA approaches and methodologies as supported by available software tools along with their applications is provided in Table. 1. Different methods are appropriate for different applications. For example, CTA is applied for determining cognitive demand and skill, whereas HTA is more suitable for risk scoring and error description.

2.2 Evaluating Performance and Potential Human Error

The security of a system is directly or indirectly dependent on human interaction [36, 34]. Thus, defining security as a socio-technical work system in progress, where humans are threat to the system. The human-centered security concerns include procedures to complete a task, authentication required in case of multiples systems, and the theft of physical systems (laptops, hard-drives etc).

The Generic Error Modelling System (GEMS) is a reference model that accounts for the socio-technical nature of work [34]. The model explains the slips (failure to complete action), lapses (forgetting something) and mistakes (unintentional violation of rules) as *Active Failures* which are caused by humans. Violations made by humans are categorised as active failures. *Latent Failures* are explained as the resident pathogens; they are the insiders who made the breaches. The system defects inherited due to poor design, faulty maintenance and poor management decisions impose a great security and safety threat to system [5].

The cognitive attributes and models are used to identify the human factors concerns and issues, as this is one of the determining factors for human performance and reliability [23]. For instance, the study in [25] offers a practicable tool for determining the cognitive attributes responsible for human performance for critical infrastructure of rail.

Previous work defines *vigilance* as the ability to remain alert for a defined period of time. Memory, attention, visual information processing abilities, auditory and visual display are identified as vigilance increment factors, as compared to multi-tasking and reading texts which are vigilance decrement factors [2]. A decision-making process that allows a user to choose best option during a given scenario is termed as *situation awareness* [17]. During task operation, the critical thinking abilities combined with workload are necessary for better situation assessment [24]. Usually, the models for human performance tend to focus on cognitive aspects of *workload* rather than physical [6], where this cognitive attribute is dependent on skills, Human Machine Interface (HMI) design, rules and guidelines [25]. On the other hand, lack of control and fear of task failure are considered *stress* inducing factors [9]. In addition, the *risk awareness* is also considered as one of the cognitive attributes and this culture is promoted by expertise, technical abilities, better communication skills and knowledge [26].

Based on Reason's error taxonomy of cognitive, behavioural, personal and organizational factors, the Human Factors Analysis and Classification System (HFACS) framework represents four levels of failures and error sources [40]. HFACS has been used by critical infrastructure stakeholders to determine the human error sources behind accidents and incidents [41]. Human factors experts use this framework to investigate the accidents by identifying and classifying the human causes in the form of errors, mistakes or violations. Ultimately, it is the job of the system design to ensure safe acts, by making certain that there is no room for any human mistakes or errors.

2.3 Usable Security and Requirements Engineering

The threat to a system in an environment is usually caused by an attacker which is the human element responsible for compromising the security [36]. Therefore, the human factors approaches are necessary but not sufficient, and need specific usable security consideration. Security engineers now give importance to human dimension of system during design phases by considering the usability attributes during asset identification, threat scenario, misuse case, task duration, responsibility modelling etc [21].

Therefore, the concept of effective information security revolves around the idea of HCI-security of the system. The HCI-security expertise takes the form of design principles and user-centered approaches for designing usable security [37]. In the following sub-sections, the secure and usable modelling techniques along with available tool-support options are discussed:

Assured Personas The term *Personas* explains the archetypical behaviour of users. This is based on ground information collected from similar environments, where the user is expected to act [10]. According to [31], the system design can be understood from an assumptive perspective. For personas, the data sources and information obtained are backed up by imagining a variety of roles in which the personas are likely to be categorised [32].

In addition to roles, personas can also be supported by stories and scenarios. A better and refined system view can be obtained by generating personas within relevant narrative scenarios and real-life situations [30]. The story-based personas have better chances of explaining the user behaviours [32]. A persona built from a user-centered design approach has better chances of being used for various analysis purposes [21] for example, threat modelling and risk analysis.

The argumentation models within personas are based on Toulmin's model of argumentation, such that each characteristic is justified by one or more *grounds* that evidence the validity, *warrants* that act as inference rules connecting the grounds to the characteristic, and *rebuttals* that act as counterarguments for the characteristic. A model qualifier is also used to describe the confidence in the validity of the characteristic [39].

These argumentation models are used to act as the source of confirmation, for data sources used as document references for designing security approaches like roles and personas definition. The document references in the form of factoids (arguments) are elicited by carefully reading the data sources, which are used to do the affinity diagramming. For this purpose, a *Trello*[5] board can be used to organise the factoids into different groups. The assumption data is organised into clusters of similar characteristics in several sessions and discussions with relevant stakeholders.

[5] https://trello.com

Use-Case The inclusion of goal and responsibility in single structural format is represented as a use-case [8]. Usually, use-case is written in the form of scenario where an actor is associated with goal leading to fulfilment of responsibility. A general template comprises of use-case name, scope, level, pre and post conditions, actions, and other characteristics enabling to consider functional requirements and scope of project [7]. The traditional use-case approach is used to write narratives for misuse cases, for identifying security requirements [38]. However, the lack of appropriate principles and guidelines for writing a use-case, makes it an approach with open-end results and solutions.

KAOS Goal Modelling Language Goal and task models can help the security engineers to better understand the system threat model. The Knowledge Acquisition in autOmated Specification (KAOS) is a method for analysing, specifying, and structuring goals required for a system [12]. The goals and tasks modelled using UML-class diagrams, may indicate security requirements that need to be fulfilled, along with possible obstacles that model obstructions to system goals.

IRIS and CAIRIS The Integrating Requirements and Information Security (IRIS) process framework [20] was devised to understand how design concepts associated with security, usability, and software engineering could be aligned. It is complemented by the Computer Aided Integration of Requirements and Information Security (CAIRIS) platform. CAIRIS acts as an exemplar for tool-support to manage and analyse design data collected when applying an IRIS process. IRIS and CAIRIS have been used in several real-world case studies, including the development of security policies for critical infrastructure systems [22].

Vulnerabilities and threats contribute to potential risks, and threats are contingent on attacker's intent. This intent helps analysts identify the tasks and goals they carry out or exploit, which can help determine human factors issues in the form of human errors (active failures). Also, the roles present personas which help stakeholders to determine the task scenarios in more detail [3]. These task scenarios can be used by human factors engineers to inform hierarchical and cognitive task analysis which can predict the reliability of systems in different environments. Also, the identification of threats/vulnerabilities/risks (risk analysis) can be orthogonal to things like TA. Consequently, although not explicitly designed with safety in mind, IRIS provides a foundation for integrating security, safety and human factors.

3 Approach

We have devised an approach based on personas for task elicitation and use-case specifications informed Task Analysis (TA). The concepts associated within this approach are shown in UML class diagram in Figure 1. The personas narrative elaborates the task performed by a role, which helps to identify tasks for analysis. Second, TA is conducted using a use-case specification pre-defined format. Finally, for each use-case specification Cognitive Task Analysis (CTA) and Hierarchical Task Analysis (HTA) is performed. CTA is conducted by scoring relevant cognitive reactions. This leads to identification of different levels of human failures with the use of *Algorithm 1*. During HTA, associations between use cases are identified. After colour coding of the use cases, graphical models are generated based on *Algorithm 2*.

The use case models with specified level of human failures help security and safety engineers better make sense of the associated risk modelling and safety analysis elements, like vulnerabilities, threats and potential hazards. Use cases with the highest level of human failures are categorised using Human Factors Analysis and Classification System (HFACS) framework to inform specific human error sources.

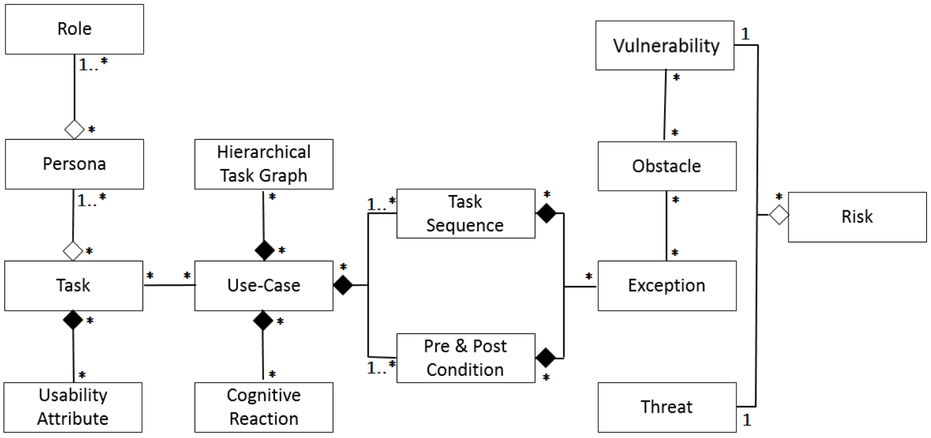


Fig. 1. Use-Case Specifications Informed Task Analysis Concepts

3.1 Personas for Task Elicitation

Personas are based on the Toulmin's Argumentation Models (Grounds, Warrants and Rebuttals), which aim at providing proper structure and assurance for qualitative data analysis [4]. This approach is automated by using tool-support at different stages, such as *Trello* board for organising factoids into clusters and CAIRIS for importing factoids and establishing persona characteristics by generating argumentations models [20]. Using these argumentation models, the personas characteristics are identified, and scenario-based narrative is written.

These personas narratives are used to elicit tasks for TA. After task elicitation, the relevant stakeholders are presented with organised information in rough tabular forms and their feedback is used to validate data for writing proper use-case specifications for conducting TA.

3.2 Use-Case Specifications Informed Task Analysis

TA is conducted for the elicited tasks. For this purpose, use-case specifications are used as data gathering, representation and analysis tool. Use cases allow both the user and functional characteristics of system to be presented, simultaneously.

As human factor engineers commonly use spreadsheets to support their workflow, we propose using Microsoft Excel (or a similar application) for managing TA data. We developed a script in Python to convert the spreadsheet data to the CAIRIS XML model format, so subsequent import into CAIRIS. A set of attributes are defined for the preparation of use-case specifications, including use case title, abbreviated title, use case id, actor/s, objective, pre and post condition/s, task sequence and exception/s. The choice for these attributes is based on two components of the system: *user* and *function*. These selected attributes simplify complexity, by making it easier for stakeholders to read, understand and analyse use cases. Finally, the use case specifications are presented to human factors experts for validation through feedback. Afterwards, these use case specifications are imported into CAIRIS.

3.3 Cognitive Task Analysis

Cognitive Task Analysis (CTA) is conducted to evaluate cognitive reactions against each use case. Previous work has shown that five cognitive reactions have an influence on human performance based on Performance Shaping Factors (PSFs), such as tiredness, emotional tension, skills, Human Machine Interface (HMI) design, rules, guidelines, and safety awareness [25]. Therefore, we use these following five cognitive reactions to evaluate human failures: i) vigilance, ii) situation awareness, iii) workload, iv) stress, and v) risk awareness. These are further described in Table 2.

Table 2. Cognitive Reactions and Performance Shaping Factors

Cognitive Reaction	***Performance Shaping Factors***
Vigilance	Tiredness, emotional stress, tension and fatigue.
Situation Awareness	Skill-set of an individual and Human Machine Interface (HMI) design.
Workload	Skills, HMI design, rules and guidelines.
Stress	HMI design, rules and guidelines.
Risk Awareness	Safety awareness, rules and guidelines.

For each use case, *Low, Medium, High or None* values are assigned to these cognitive reactions based on expert rationale. To collect the values and rationale, open-ended semi-structured interviews are held with relevant stakeholders. There is no mandatory list of questions, but the intent is to elicit knowledge through on open discussion. The stakeholders are presented with the proposed use case specifications, where they are asked to select different values for cognitive reactions, and document and justify their rationale.

Using values of cognitive reactions stored in CAIRIS, *Algorithm 1* determines different levels of human failures. Each use case is taken as an input, and provides level of human failure for that specific use case as output. For each use case, *cognitive_reaction[n]* returns an array of 5 values of cognitive reactions where n ranges from 1 to 5. The values of *cognitive_reaction[n]* vary from *High, Medium, Low or Null.* The values are also associated with numbers such as, *(0 for Null, 1 for Low, 2 for Medium and 3 for High).* The mean (ranging from 0 to 3) of these cognitive reactions is calculated to determine different levels of human failures. Mean is a suitable measure of central tendency, as median only points out the middle value while ignoring the individual value behind each cognitive reaction, and mode determines extreme values either too high or too low. There are three levels of human failures against mean, *0 or 1 for Low, 2 for Medium and 3 for High*, where *Low* being the use-case with less chances of human failure and *High* being the use-case with extreme chances of human failure.

Algorithm 1: Level of Human Failure for each Use-Case

```
Data: $u$ - the use-case specification
Result: $l$ - the level of human failure for $u$
Function failurelevel(u) is
    $sum = 0$;
    for $n \leftarrow 1$ to 5 do
        $sum$ += cognitive_reaction[n];
    $mean \leftarrow$ round($sum/5$);
    if $mean <= 1$ then
        $l \leftarrow$ Low;
        break;
    if $mean == 2$ then
        $l \leftarrow$ Medium;
        break;
    if $mean == 3$ then
        $l \leftarrow$ High;
        break;
```

3.4 Hierarchical Task Analysis

The task hierarchy is drawn from the task sequences as stated in use case specifications. The high-level use cases and tasks are divided into low-level use cases and tasks, where each use case is filled in with a particular colour depending on level of human failure assigned to it. The colour mapping is *dark blue, blue and light blue* for *High, Medium and Low* level of human failure, respectively. Using these colour codes, the different levels of human failures are better illustrated with HTA graphs using *Algorithm 2.* These different levels of human failures highlight use cases and tasks requiring more attention by human factors, safety and security experts for design analysis.

The *Algorithm 2* takes no input instead its output is a set of quadruples i.e., (h, h_fl, t, t_fl) in which h is the head task name, h_fl is the head task failure level, t is the tail task name, and t_fl is the tail task failure level. The empty sets are defined for the quadruples hta, and task node/failure level pairs *visited* while enumerating the set (lines 2 & 3). The *buildTaskGraph* is a function that generates a set of tuples from the CAIRIS model. Using this function, the algorithm retrieves a set of tuples (h,t) in which h is the head task name and t is the tail task name. Each tuple in *buildTaskGraph* is enumerated, if h intersects with the first element in visited set then the task node/failure level from the set is retrieved (lines 6 & 7), otherwise the *failurelevel* using *Algorithm 1* is calculated for the task node and *union* of task node/failure level with *visited* set is done (lines 9 & 10). These steps are repeated for t (lines 12-17). Once we have tuples for h and t then quadruple is constructed by performing *union* with quadruple set *hta* (line 18). On completion of the algorithm, quadruple set is returned (line 20).

Algorithm 2: Build HTA Graph

```
   Input  : None
   Data: $tg$ - set where each element is a tuple ($h$,$t$) in which $h$ is the head task name and $t$
         is the tail task name, $tt$ - tuple drawn from $tg$, $visited$ - set where each element is a
         tuple ($t$,$fl$) in which $t$ is the task name, and $fl$ is the task failure level, $h\_fl$ - tuple
         ($h$, $fl$) where $h$ is the head task name and $fl$ is the head task failure level, $t\_fl$ -
         tuple ($t$,$fl$) where $t$ is the tail task name and $fl$ is the tail task failure level.
   Output: $hta$ - set where each element is quadruple ($h$, $h\_fl$, $t$, $t\_fl$) in which $h$ is the
           head task name, $h\_fl$ is the head task failure level, $t$ is the tail task name, and
           $t\_fl$ is the tail task failure level.
 1 Function buildHTAModel is
 2     $hta \leftarrow \emptyset$;
 3     $visited \leftarrow \emptyset$;
 4     $tg \leftarrow$ buildTaskGraph;
 5     while $tt \leftarrow tg$ do
 6         if $tt[0] \in visited$ then
 7             $(h, fl) \leftarrow visited\ tt[0]$;
 8         else
 9             $(h, fl) \leftarrow failurelevel\ (tt[0])$;
10             $visited \leftarrow visited \cup (h,fl)$;
11         end
12         if $tt[1] \in visited$ then
13             $(t,fl) \leftarrow visited\ tt[1]$;
14         else
15             $(t,fl) \leftarrow failurelevel\ (tt[1])$;
16             $visited \leftarrow visited \cup (t,fl)$;
17         end
18         $hta \leftarrow hta \cup (h, h\_fl, t, t\_fl)$;
19     end
20     return $hta$;
21 end
```

3.5 Risk Analysis

Within the use case specification, an exception is an undesirable situation where the task sequence is disturbed. The security and safety experts are given the opportunity to analyse exceptions in detail for the possibility of potential vulnerabilities, threats, risks and hazards during tasks within system. As with the exploitation of vulnerability, the risk of occurrence of threat may lead to catastrophic accident due to potential hazard. Therefore, by using the human factors approach of task analysis, the identified exceptions within use case specifications help to achieve safe and secure design solutions by risk analysis.

3.6 Implementation in CAIRIS

For demonstrate how this approach can be tool-supported, we have forked the GitHub repository of CAIRIS and implemented *Algorithm 1* and *2*. The forked GitHub repository is available at link: *https://github.com/s5121191/cairis* and can be reviewed for implementation details.

4 Preliminary Evaluation: Identifying Tasks for Human Error Potential

Due to technological advancements in rail infrastructure, many operational tasks are becoming more centred around mental (cognitive) abilities rather than physical. Following the deployment of the European Railway Traffic Management System (ERTMS), working relationships are more dependent than ever on team coordination capabilities. For example, the driver and signaller work in conjunction with each other to ensure safe and efficient operations. Mindful of this, we used the ERTMS specifications [35] to conduct a Task Analysis (TA) of the role of *Train Signaller*[6]. We have sketched a rough profile of *A Day in the Life of a Train Signaller*, which consisted of task breakdown in a time-line from 0030 to 2350 hours.

Table 3. Documentation and Literature used for Train Signaller Personas

Ser.	***Article Title***	***Author***	***Publisher***
1.	A Day in the Life of a Train - Operational Concept [18]	ERTMS	Operational Principles and Rules - Technical Document
2.	Network Rail - Signalling Control Centers [33]	Network Rail	Published and Issued by Network Rail - Module A5-5
3.	Operational Concept for The European Railway Traffic Management System [35]	Rail Safety and Standards Board	RSSB-ERTMS-OC Issue 2
4.	Understanding Railway Signaller Tasks and Operations [28]	Ex-Signalman and Human Factors Consultant	Interview Notes

4.1 Personas for Task Elicitation

The ERTMS Operational Concept was used to develop an understanding of the job of Train Signaller. The open-source documentation and literature specified in Table 3 was used to ground our knowledge. We supplemented this knowledge by interviewing a number of other relevant rail stakeholders. A total of 4 interviews were conducted, one from human factors expert with focus on TA methodologies, one from safety engineer for potential hazard analysis using human-error sources

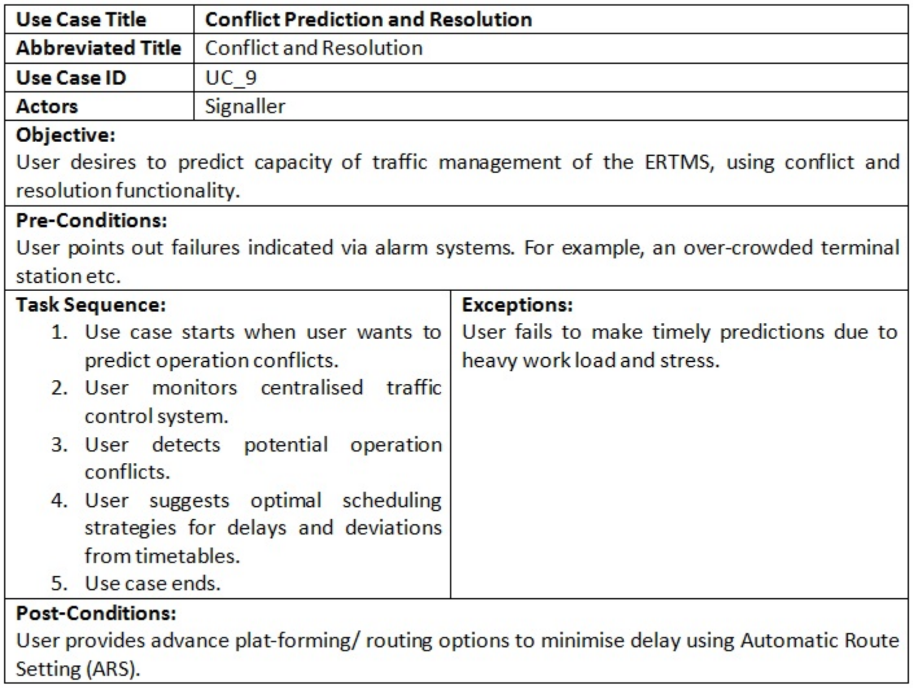
[6] The complete CAIRIS model of this analysis is available at *https://github.com/s5121191/CRITIS-21*.

and two from train signallers for collecting data about ERTMS signalling tasks performed in routine.

We defined models associated with the role of rail *Signaller*. From our knowledge base, we elicited 73 factoids, which grounded 11 argumentation models for the persona of a train signaller (*Daniel*). These argumentation models contributed towards the narrative of Daniel, explaining his activities, attitudes and aptitudes. Using personas narrative for *Daniel*, 16 major tasks were elicited for the role of train signaller. For example, the task of *Combine Workstation* is found from persona characteristic of activities for *Daniel* as shown by highlighted text.

> *Daniel is performing the job of railway signaller. Daniel working from his signaller's workstation is responsible for monitoring and controlling train movements after* ***combining workstations****.*

These tasks were organised in rough tabular form and fed back to stakeholders for validation.



Use Case Title	**Conflict Prediction and Resolution**
Abbreviated Title	Conflict and Resolution
Use Case ID	UC_9
Actors	Signaller
Objective: User desires to predict capacity of traffic management of the ERTMS, using conflict and resolution functionality.	
Pre-Conditions: User points out failures indicated via alarm systems. For example, an over-crowded terminal station etc.	
Task Sequence: 1. Use case starts when user wants to predict operation conflicts. 2. User monitors centralised traffic control system. 3. User detects potential operation conflicts. 4. User suggests optimal scheduling strategies for delays and deviations from timetables. 5. Use case ends.	**Exceptions:** User fails to make timely predictions due to heavy work load and stress.
Post-Conditions: User provides advance plat-forming/ routing options to minimise delay using Automatic Route Setting (ARS).	

Fig. 2. Use-Case Specification for 'Conflict Prediction and Resolution'

4.2 Use Case Specifications Informed Task Analysis

Use cases were identified and specified, using a pre-defined format. Using *Microsoft Excel*, points were scribbled down along-side data collection. This was an iterative process, where each use case specification went through series of transformations. There were three major parts for each use-case: actor (performing the task), steps (task sequence) and conditions (identifying constraints/ exceptions). After careful consideration, a total of 16 use case specifications were specified. For example, Figure 2 specifies a use case for *Conflict Prediction and Resolution*. Following validation from stakeholders, these use case specifications were imported into CAIRIS.

4.3 Cognitive Task Analysis

After specifying the use cases, CTA was conducted by scoring each use case against cognitive reactions. For example, in the use-case of *Conflict Prediction and Resolution*, the values assigned were as follows: vigilance was *High*, situation awareness was *Medium*, workload was *High*, stress was *High* and risk awareness was *Medium*, with a defined rationale where *under manual control train movements or alterations in timetable may cause additional workload*. These values of cognitive reactions were fed into the *Algorithm 1*, where the mean was evaluated as *3*. This indicated that the *Conflict Prediction and Resolution* use case was associated with a *High* level of human failure.

Table 4. Cognitive Task Analysis for Use-Case Specifications

Use-Case ID	***Use-Case Name***	***Vigilance***	***Situation Awareness***	***Workload***	***Stress***	***Risk Awareness***	***Level of Human Failure***
UC-1	Combine Workstation	Low	High	Medium	Null	High	Medium
UC-2	Grant Possessions and Isolation	Medium	Medium	Low	Null	Medium	Low
UC-3	Maintain Operations Log	Low	Medium	Low	Low	Low	Low
UC-4	Ensure Normal Service Delivery	Low	Medium	Medium	Low	Low	Low
UC-5	Monitor Regulator Intervention	Low	Low	Medium	Medium	Low	Low
UC-6	Conduct Manual Routing	High	Low	Medium	Low	High	Medium
UC-7	Plan Stock Positioning	Low	Low	Medium	Low	Low	Low
UC-8	Grant Off-Peak Blockage	High	Medium	Medium	High	High	High
UC-9	Conflict Predict and Resolution	High	Medium	High	High	Medium	High
UC-10	Issue Temporary Timetable	Medium	Medium	Medium	Low	High	Medium
UC-11	Identify Broken Rail	High	Low	Low	Medium	Medium	Medium
UC-12	Test Back-up Facilities	Medium	Medium	Low	Low	Low	Low
UC-13	Map Operational Planning	High	Medium	Medium	Low	High	Medium
UC-14	Run Route Availability	High	High	Medium	High	Low	Medium
UC-15	Run Sectional Time	Low	Low	Low	Low	Low	Low
UC-16	Order of Implementation	Medium	Medium	Low	Low	Low	Low

Consequently, the design analysis of this use case lead to situations where there is a strong tendency towards mistakes or errors. The different values of cognitive reactions for the use cases is shown in Table 4, together with with mean calculation from *Algorithm 1*.

4.4 Hierarchical Task Analysis

With the help of *Algorithm 2*, the colour coded HTA graph was generated with use cases of Low, Medium or High level of human failures, as shown in Figure

3. Here, in the HTA graph, 9 use cases and tasks can be seen impacting each other. Based on the full HTA graph, 3 use cases – *Combine Workstations, Grant Off-Peak Blockage* and *Conflict Prediction and Resolution* – correspond with *High* levels of human failure.

By conducting TA as a combination of CTA and HTA tools, the cognitive load on humans parallel to hierarchy of tasks is better understood. For example,the use case *Map Operational Planning* depends on *Run Route Availability* and *Run Sectional Time*, where cognitive reactions like *vigilance, situation awareness and workload* are important. This breakdown highlights tasks dependency and logic behind goals, whereas resources, time and expertise are evaluated using cognitive reactions. Both of equip human factors experts with sufficient knowledge when making design decisions.

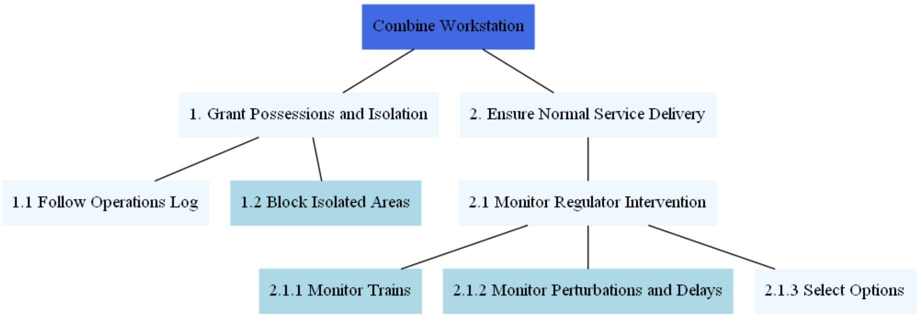

Combine Workstation
1. Grant Possessions and Isolation
2. Ensure Normal Service Delivery
1.1 Follow Operations Log
1.2 Block Isolated Areas
2.1 Monitor Regulator Intervention
2.1.1 Monitor Trains
2.1.2 Monitor Perturbations and Delays
2.1.3 Select Options

Fig. 3. HTA Graph with Levels of Human Failure

4.5 Risk Analysis

During the specification of *Conflict Prediction and Resolution*, an exception was identified where a user fails to make timely predictions due to heavy workload and stress. This might occur due to the vulnerability of *Lack of Independent Check*, where the user should update checklists with timely prediction data. This vulnerability affords two threats: *Delays during Routing* and *Operational Conflicts*. These threats contribute to the risk *Failure of Automatic Route Settings*, and this failure leads to hazard *Collision between Trains*, with severe consequences.

5 Discussion

The approach entails TA as the human factors technique for determining potential human error sources. These human error sources highlight possible security risk elements in the form of vulnerability, threat, risk and hazard. The intent and effort are recognised using CTA, where attributes like vigilance, situation awareness, workload, stress and risk awareness are contributing factors. However, the task and use case hierarchical breakdown using HTA contributes to an understanding the division of effort between these tasks.

This is an area where human factors experts can provide important feedback. When presented with human error sources behind tasks performed, they can benefit from a graphical visualisation to show the tasks requiring more attention. By collaborating with security and safety engineers, the potential hazards arising from these tasks are also visualised using threat modelling and risk analysis in CAIRIS. Here, CAIRIS developed the link between tasks identified for TA and vulnerabilities resulting from these tasks.

With the occurrence of exceptions, possible exploitation opportunities were identified. For example, in our case study, the major exceptions found in the use cases are *power failure*, *equipment failure*, *conflicts and delays*, *track circuit failure*, etc. These exceptions link to KAOS goal models, which give security and safety experts an idea about possible vulnerabilities leading to threats, risks and hazards (i.e. risk analysis). Similarly, the cognitive reactions defined against each use case could determine the potential human error sources using HFACS framework. Using HFACS, each use case with the highest level of human failure is labelled against the closest possible description of human error. For example, the use case *Conflict Prediction and Resolution* corresponds to a high level of human failure, where vigilance, workload and stress are important. Hence, the chances of occurrence of *Skill-based Error* and *Violation* are high, requiring scrutiny from human factors experts. Vigilance and workload may lead to the identification of *Decision Error*, but this is unlikely because this type of error results from a wrong judgement during emergency situations, rather than during routine operations.

6 Conclusion

In this paper, we present an approach where use cases drive TA for designing and evaluating safe and secure rail infrastructures. We catalogue the rail infrastructure for design analysis and, through a preliminary evaluation on regular tasks performed by an ERTMS Signaller, highlight human error sources behind these tasks. In doing so, we show how these human error sources contribute towards design solutions by identifying safety hazards and security risks.

In presenting our approach, we have made three contributions. First, we have derived a TA approach from the security and requirements engineering IRIS framework using concepts such as roles and personas, task and goal-obstacle modelling. Second, we have shown how CTA and HTA can be combined as single, tool-support TA approach to highlight the importance of mental load with a detailed task breakdown. Finally, we have shown how use case specifications assist with task sequencing and exception identification. These exceptions help security and safety experts to conduct risk and hazards analysis by identifying potential vulnerabilities and threats hidden beneath system design.

TA with CAIRIS as tool-support facilitates other kinds of analysis, including asset, goal-obstacle, responsibility, threat and risk modelling, and even hazard investigation using safety analysis techniques. Thus, by using this approach the human factors experts are given a chance to work in collaboration with security and safety experts to analyse and make collective design decisions in critical infrastructure. As future work, we build on our approach by integrating further human factor techniques and methods to further facilitate the design of safe, secure, and usable rail solutions.

Acknowledgements

The work described in this paper was funded by the BU studentship *Integrating Safety, Security, and Human Factors Engineering in Rail Infrastructure Design & Evaluation*. We are also grateful to Ricardo for their support.

References

1. Affairs, A.S.f.P.: Task Analysis. /how-to-and-tools/methods/task-analysis.html (Sep 2013)
2. Al-Shargie, F., Tariq, U., Mir, H., Alawar, H., Babiloni, F., Al-Nashash, H.: Vigilance Decrement and Enhancement Techniques: A Review. Brain Sciences **9**(8) (Jul 2019)
3. Altaf, A., Faily, S., Dogan, H., Mylonas, A., Thron, E.: Identifying Safety and Human Factors Issues in Rail using IRIS and CAIRIS. In: CyberICPS 2019: 5th Workshop On The Security Of Industrial Control Systems & Of Cyber-Physical Systems. Springer, Luxembourg, Luxembourg (Sep 2019)
4. Atzeni, A., Cameroni, C., Faily, S., Lyle, J., Flechais, I.: Here's Johnny: A Methodology for Developing Attacker Personas. In: 2011 Sixth International Conference on Availability, Reliability and Security. pp. 722–727. IEEE, Vienna, Austria (Aug 2011)
5. Brostoff, S., Sasse, A.: Safe and Sound: A Safety-Critical Approach to Security p. 10 (2001)
6. Cao, S., Liu, Y.: Modelling workload in cognitive and concurrent tasks with time stress using an integrated cognitive architecture. International Journal of Human Factors Modelling and Simulation **5**, 113 (Jan 2015)
7. Cockburn, A.: Basic Use Case Template (2), 8 (26-October- 1998)
8. Cockburn, A., Bank, N.: Structuring Use cases with goals (Dec 1997)
9. Conway, D., Dick, I., Li, Z., Wang, Y., Chen, F.: The Effect of Stress on Cognitive Load Measurement. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) Human-Computer Interaction – INTERACT 2013. pp. 659–666. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2013)
10. Cooper, A.: The Inmates Are Running the Asylum. Macmillan Publishing Co. (1999)
11. Crandall, B., Klein, G., Hoffman, R.R.: Working Minds: A Practitioner's Guide to Cognitive Task Analysis. MIT Press, Cambridge, MA (2006)
12. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming **20**(1), 3–50 (Apr 1993)

13. Davis, W., Burton, A.: Ecological Task Analysis: Translating Movement Behavior Theory into Practice. Adapted Physical Activity Quarterly **8**, 154–177 (Apr 1991)
14. Diaper, D., Stanton, N.: The Handbook of Task Analysis for Human-Computer Interaction. CRC Press, Mahwah, NJ (2004)
15. Embrey, D.: Task Analysis Techniques p. 14 (2000)
16. Embrey, D.D., Zaed, S.: A Set Of Computer Based Tools Identifying And Preventing Human Error In Plant Operations p. 11 (2021)
17. Erbacher, R.F., Frincke, D.A., Wong, P.C., Moody, S., Fink, G.: A Multi-Phase Network Situational Awareness Cognitive Task Analysis:. Information Visualization (Jan 2012)
18. ERTMS: A Day in the Life of a Train - Operational Concept (Apr 2019)
19. European Network and Information Security Agency: Railway Cybersecurity: Security Measures in the Railway Transport Sector. Publications Office, LU (2020)
20. Faily, S.: Designing Usable and Secure Software with IRIS and CAIRIS. Springer International Publishing, Cham (2018)
21. Faily, S., Fléchais, I.: Barry is not the weakest link: Eliciting Secure System Requirements with Personas p. 8 (Sep 2010)
22. Faily, S., Flechais, I.: User-Centered Information Security Policy Development in a Post-Stuxnet World. In: 2011 Sixth International Conference on Availability, Reliability and Security. pp. 716–721. IEEE, Vienna, Austria (Aug 2011)
23. Felice, F.D., Petrillo, A.: Methodological Approach for Performing Human Reliability and Error Analysis in Railway Transportation System p. 13 (2011)
24. Golightly, D., Balfe, N., Sharples, S., Lowe, E.: Measuring situation awareness in rail signaling. In: Rail Human Factors Around the World: Impacts on and of People for Successful Rail Operations. pp. 361–369 (Apr 2009)
25. Hammerl, M., Vanderhaegen, F.: Human Factors In The Railway System Safety Analysis Process: 3rd International Rail Human Factors Conference p. 9 (2009)
26. Jen, R.: How to increase risk awareness. In: PMI® Global Congress 2012. PA: Project Management Institute., Vancouver, British Columbia, Canada, North America (2012)
27. Jonsson, E., Olovsson, T.: On the Integration of Security and Dependability in Computer Systems p. 6 (1998)
28. Martin, K.: Understanding Railway Signaller Tasks and Operations (Feb 2020)
29. Militello, L., Hutton, R.: Applied Cognitive Task Analysis (ACTA): A Practitioner's Toolkit for Understanding Cognitive Task Demands. Ergonomics **41**, 1618–41 (Dec 1998)
30. Nielsen, L.: Personas - User Focused Design. Human–Computer Interaction Series, Springer-Verlag, London (2013)
31. Norman, D.: Emotional Design: Why We Love (or Hate) Everyday Things. Basic Books (2004)
32. Pruitt, J., Grudin, J.: Personas: Practice and Theory. In: Proceedings of the 2003 Conference on Designing for User Experiences. pp. 1–15. DUX '03, ACM, New York, NY, USA (2003)
33. Rail, N.: Network Rail - Signalling Control Centers (Jun 2018)
34. Reason, J.: Human Error by James Reason (Oct 1990)
35. RSSB: Operational Concept for ERTMS (Jun 2014)
36. Schneier, B.: Secrets and Lies: Digital Security in a Networked World. John Wiley & Sons (2000)
37. Shostack, A.: Threat Modeling: Designing for Security. John Wiley and Sons, Indianapolis, IN (2014), adam Shostack

38. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements Engineering **10**(1), 34–44 (Jan 2005)
39. Toulmin, S.E.: The Uses of Argument p. 259 (2003)
40. Wiegmann, D.A., Shappell, S.A.: A Human Error Approach to Aviation Accident Analysis: The Human Factors Analysis and Classification System. Routledge, Aldershot, Hants, England ; Burlington, VT, 1 edition edn. (Jul 2003)
41. Zhou, J.L., Lei, Y.: Paths between latent and active errors: Analysis of 407 railway accidents/incidents' causes in China. Safety Science **110**, 47–58 (Dec 2018)