Human Orientedness of System Safety

Ray Feodoroff

https://orcid.org/0000-0002-5518-5570

Abstract

This paper is a research proposal, really looking at whether there is impetus to investigate a harmonising of ideas across several research areas. Ultimately this is to investigate the best means of weaving assurance reasoning into graphical requirements and design notations. The basis for this approach is a recurring emphasis on goal-orientation over the last 20 or more years of the class Leveson has described as "human centric". This paper is a first step, to look for the theoretical basis for that graphical approach.[1.]

Keywords: problem-oriented, goal-oriented, agent-oriented, means-directed, human-centered.

1 Introduction

Cassano et al. (Cassano, et al., 2021) have previously discussed their believe that safety reasoning is not case-based and hazard-oriented as is the current custom, but might better involve reasoning about actions, modality, and agency. This begs the question therefore: is safety reasoning better addressed through ideas that try to understand socio technical systems by seeing the world through anthropomorphized viewpoints?

Ideas, therefore, from goal-oriented requirements engineering (Yu, 1995; van Lamsweerde, 2009), and from problem-oriented or (so-called) Assurance Driven Design (ADD) (Hall & Rapanotti, 2009). Ideas also from Intent Specifications and STPA (System Theoretic Process Analysis) as input to same (Leveson, 2003).

The aim is to eventually design a graphical notation to support design assurance reasoning directly in the requirements/design notation. The second step, towards designing a graphical notation, according to Hall, Rapanotti and Jackson is to develop an intermediate non-interpreted Domain and Requirement Description Language (DRDL) (Hall, et al., 2005). Once the DRDL is defined, we may subsequently move onto the third and final step of designing of a graphical notation.

A DRDL is otherwise, according to Hall et al. (Hall, et al., 2005), a language or a collection of languages to describe domains, phenomenon, requirements and specifications. The notion of a collection of languages is, we feel, important as each layer of any decomposition, over an abstraction hierarchy, may indeed have a different language. Though each layer of language needs to relate to the one above and below to facilitate the decomposition process.

Principally, however, before setting to work on a DRDL in subsequent research, we need look at the first step, which is to define the theoretical basis for the DRDL and hence the discussion herein. In the process of defining the theoretical basis for our DRDL, we will transform the problem-oriented ideas into a goal-oriented style, while maintaining the importance of problem decomposition.

The basis for this research is otherwise to leverage ideas out of cognitive science and cognitive system engineering and how anthropomorphism of the world has always appeared to be in the background of human consciousness, through notions of agency. Closely related work is, of course, Agent Oriented software engineering practices (Abdalla & Mishra, 2021), which are based somewhat on Bratman's book "Intention, Plans, and Practical Reasoning" (Bratman, 1999).

1.1 Lip Service to Rasmussen

Jens Rasmussen was a researcher in areas such as human factors, cognitive systems engineering, and in system safety. Leveson has detailed the influence of Rasmussen on both the notion of Intent Specifications, and the use of control-theoretics behind the notion of the System Theoretics of STPA (Leveson, 2017). System Theoretics essentially being the observation of the control-theoretic loops that should exist between elements of a system to maintain control of that system, both physically and organisationally. Any paradigm of change was, however, only because there appeared to be two streams for system safety, a cognitive system engineering-based stream and another stream. There is still a question as to whether there should be two system safety streams, if pulling cognitive system engineering concepts into another stream is an important paradigm shift for that other stream.

Given the cognitive system engineering aspect, we note therefore, Leveson's position that beliefs held by the operator in the state of the system, at any point in time, play a key role in that control-theoretic model. Especially where the human operator acts as a controller at any level in the system, or indeed the enterprise. The beliefs of the designer, during the act of design, similarly appear important as STPA attempts to methodically drive the design through discovery of both wanted and unwanted control actions, actions discovered by beliefs the designer has in the controller's internal processes and the potential errors in those processes. Of course, if we are thinking about the human as a controller, we might refer to those internal processes not as computational but as

[1] This paper appeared at the Australian System Safety Conference (ASSC2023), Adelaide, Australia.

deliberations, since we are not animal "machines" per René Descartes[2], we are humans.

Certainly, a deliberating agent's goals, beliefs and plans are first-class elements in an agency-based model (Bratman, 1999), whereas they are not yet in a control-theoretic model per se (Thomas, 2013). More discussion on this under Section 2.

Leveson also explains the influence on the hierarchy of an Intent Specification by the notion of Abstraction Hierarchy, which is essentially one means to decompose a system by a set of related abstractions (Leveson, 2003). That use of decomposition being a means to ensure the user's goals for the system are driven into the system design, though the act of decomposition is managed by document sections, as opposed to a model-based approach.

1.2 Decomposition as an Assurance driver

Decomposition of case arguments, using notations such as GSN, aims to document the asserted chain of reasoning over the argument graph. So, we can agree that reasoning over a graph is a goal for design assurance reasoning and therefore of our DRDL. A pity then that designers do not embed assurance reasoning semantics into the more popular requirements engineering and design notations.

Analogously, Intent Specifications aim to decompose goals using means-ends thinking, with Leveson explaining Rasmussen's abstraction hierarchy concept (Leveson, 2003) with Explanation 1 below.

> *In a means-end abstraction, each level represents a different model of the same system. At any point in the hierarchy, the information at one level acts as the goals (the ends) with respect to the model at the next lower level (the means).*
>
> Explanation 1

The notion of means-ends is nuanced here. Any goal-directed approach, say in AI, would be from a current state, with a path calculated towards the next state, with each application of means eventually driving towards the end goal. The problem of decomposition, however, might be considered best a means-directed activity. This is since we start at the goal and work top-down.

This means-directed activity is required since goal decomposition aims to ensure that the quality of the act of definition of the means guarantees (somehow) that application of the means provides satisfaction of, at least, the goal immediately above. Dewey[3] called this quality "end-in-view".

Assurance justification (proof of satisfaction of the top claim) might thus be an "end-in-view" chain, from the lowest point on the graph, to the top goal (if assurance is a goal-directed outcome). With each means provided action ending in local "end-in-view" satisfaction that contributes to a monotonic non-decreasing function that acts towards proof of satisfaction of the top goal. That is, each local "end-in-view" itself must aim, piece-wise, to satisfy the overall top goal. Stated another way, the means only has value if the consequence of the action, supplied by the means, can be shown to sit on that monotonic non-decreasing function that acts towards proof of satisfaction of the top goal. This idea, that monotonic non-decreasing functions over a graph provides proof of satisfaction of a top goal is evident already in a certain class of requirements engineering (Yu, 1995; Giorgini, et al., 2003), whereas it is relatively new to case-based reasoning (Rushby, 2021).

Leveson, further explaining abstraction hierarchy (Leveson, 2003) with Explanation 2 below.

> *A change of level involves both a shift in concepts and in the representation structure, as well as a change in the information suitable to characterise the state of the function or operation at the various levels.*
>
> Explanation 2

Decomposition may therefore be a tool for design assurance, if it aims to guarantee the path over the decomposition graph is a contiguous union of disjoint expression of means, and of the claims for the quality of the means, with each change in level with a commensurate refinement of the preceding concepts and information. It is certainly the aim Leveson is hinting at for (so called) goal-oriented Intent Specifications (Leveson, 2003). A pity then that Leveson does not opt to use a suitable goal-oriented graphical reasoning notation, or a problem-oriented reasoning notation for that matter.

To understand the basis for the problem-oriented ADD approach (Hall & Rapanotti, 2009), when decomposing any multi-agent system's ***User*** requirements *R* into ***Machine*** specifications *S*, we need start with the canonical expression of the "problem frame" problem-oriented idea of Jackson (Jackson, 2000) at Eq (1), noting *P* for Problem is often omitted.

$$P: W, S \vdash R \qquad (1)$$

Eq (1), comprises the World (*W*) which constrains the Specification (*S*) to be a bounded decomposition solution for the Requirement (*R*) (the problem to be solved). Of course, if *W* were to change or be poorly documented, then *S* would no longer support entailment of *R*. This sensitivity to *W* relates to the notion of intended use in safety arguments, since we note that we often rely upon the world to be well behaved. We are otherwise saying that *S* is the means, in the context of *W*, to meet the end *R*.

The problem-oriented decomposition approach which has important properties, see Eq (2), appears in work by Hall, Mannering and Rappanotti (Hall, et al., 2007). In this case the requirement decomposition, to specification, is both right-to-left and from bottom-to-top. According to Hall et al Each expression in the top layer can be recursively decomposed upwards until a solution is reached, when no further decomposition is necessary.

$$\frac{W_1, S_1 \vdash R_1;\ \ldots; W_n, S_n \vdash R_n}{W, S \vdash R} \qquad (2)$$

The multiple layers of Eq (2), by Hall et al. (Hall, et al., 2007), relates to Explanation 1 and that difference is captured by decomposing *R* to *S*, and similarly *S* to $S_{1..n}$, and *R* to $R_{1..n}$. To constrain this, to guarantee the path over the decomposition graph is a contiguous union of disjoint

[2] https://plato.stanford.edu/entries/descartes/

[3] https://plato.stanford.edu/entries/dewey-moral/

expression of means, Hall et al. also decompose *W* to $W_{1..n}$. The top layer is therefore the means to meet the ends defined on the bottom layer in the "end in view" sense. Hall et al. proposed that Eq (2) can include a justification for the decompositions with a variant Eq (3)[4] (Hall, et al., 2007) (Hall & Rapanotti, 2009). According to Hall et al. (Hall, et al., 2007), in Eq (3), *NAME* is the name of the problem Justification. While *J* itself is the justification that provides the proof of satisfaction of the end *R*.

$$\frac{W_1, S_1 \vdash R_1;\ \dots;\ W_n, S_n \vdash R_n}{W, S \vdash R} \quad \begin{matrix}[NAME]\\ \langle\!\langle J \rangle\!\rangle\end{matrix} \qquad (3)$$

Hall et al. define *J* in the form Eq (4) (Hall, et al., 2007). With *J* described by Hall et al. as a function that accumulates "Adequacy Arguments" (*CA*), for each means-directed decision. To maintain the nature of the natural deductive sense of Eq (4), we will rewrite Eq (4) as Eq (5), to allow re-writing Eq (3) as Eq (6).

$$(CA_1 \wedge \dots \wedge CA_n) \wedge J \qquad (4)$$

$$CA_1, \dots, CA_n \vdash J \qquad (5)$$

$$\frac{W_1, CA_1, S_1 \vdash R_1;\ \dots;\ W_n, CA_n, S_n \vdash R_n}{W, J, S \vdash R} \qquad (6)$$

There are obfuscated ideas behind the notation used in Eqs (1)..(6). Notably, the approach is inherently agent-oriented since *R* is assigned to a ***User***, and *S* is assigned to a ***Machine***. This invites the idea of agent actions when Hall et al. annotate the specification *S* with phenomena observed (*o*) and phenomena controlled (*c*) as in Eq (7) (Hall, et al., 2007).

$$S_o^c \qquad (7)$$

Hence we can also write Eq (1), taking *P* for granted, as: $W, S_o^c \vdash R$. So, if we take note of the use of the term "controller" in STPA based approaches (potentially meaning either a ***User*** or a ***Machine***), then Eq (7) is analogous to the definition of "requirement functions" by Thomas (Thomas, 2013). Those STPA based "requirement functions" will be denoted herein as Ʀ and comprise a tuple of controller (*sc*), controller action (*ca*) and context (*co*), or: Ʀ $\triangleq$ ‹sc, ca, co›.

Usefully, Eq (7) is control-theoretic in the sense STPA adopts, since Ʀ's *sc* can really be either an ‹*actor*› or an ‹*agent*›. We acknowledge Thomas uses Ʀ as either of the problem-oriented *R* for ***User*** or *S* for ***Machine***, whereas Hall et al. would only assign a ***Machine*** or *S*, because of limits on the decorations in Eq (7). If it helps, assume Eq (7) is operationalized by the problem-oriented ***Machine***. Thus, within our STPA derived Ʀ tuple we map in *S* and its decorations thus: ‹*sc(S)*,*ca(c)*,*co(o)*›. In that case, *ca* acts upon the controlled phenomenon *c*, when the observed phenomenon *o* meets criteria set in the context *co*.

There are problems thinking about the problem-oriented *R* this way, for reasons related to the style Hall et al. adopt. Since, *R* is not decorated the same way as *S* (Hall, et al., 2007). We may or may not relax this in our approach for our DRDL, but we will not otherwise dwell on the decorations for *R* this pass. The difference is currently something ⟪*stereotyping*⟫ can likely address, since the decorations for *R* are a somewhat goal-based intention to "constrain" *x* with "reference" to *y*. So, that idea may be better thought of as a constraint towards *x* or, alternatively, a constraint away from the antithesis of *x* (a.k.a. avoid the anti-goal $\bar{x}$, a.k.a. be "safe"). That is, the *antecedent* for, or the means toward meeting *R* either needs to provide the means to satisfy the goal *x* or, alternatively, the means to dissatisfy the anti-goal $\bar{x}$. Hence, the discussion under Section 1.3.

As previously mentioned, in contrast to Hall et al. (Hall, et al., 2007), Thomas (Thomas, 2013) is really using Ʀ in the generic sense (again, really as either *R* and ***User*** level or *S* and ***Machine*** level). Thus, Thomas is essentially using Ʀ as would Hollnagel "F" for functions, by using the notion of "scale invariance" (Hollnagel, 2012). That is, Thomas is using Ʀ outside of the concept of an abstraction hierarchy and therefore not necessarily in the sense of needing to use a means-directed decomposition style required for assurance argumentation, at least in the manner proposed herein. Thomas, in fact, does not advocate a decomposition style for STPA as a technique. Thomas does recognise, however, that other authors have experimented with a decomposition styled approach for STPA as a technique.

Recalling that the tuple representing Ʀ (Thomas, 2013), is ‹*sc*,*ca*,*co*›, means we can define the *a priori* context to be ‹*sc(S)*,*ca(?)*,*co(o)*›. This *a priori* context is the context required to trigger *ca*, but prior to the action *ca* being invoked due to the necessity for a deliberation/computational cycle within the controller (aka ‹*agent*› or ‹*actor*›). Thus the *a priori* context ‹*sc(S)*,*?*,*co(o)*› is equivalent to Eq (8) in problem-oriented terms.

$$W, S_o^? \qquad (8)$$

So, the perceived context *W*,S_o (a.k.a. ‹_,_,*co(o)*› from Ʀ), results in a "belief" about the world *W*. In problem-oriented terms, this "belief" is based upon observed phenomena *o*, compared to an internalized model of $W_{internal}$ within the controller, prior to selection of the control action *c* (a.k.a. ‹_,*ca(c)*,_› from Ʀ). The invocation of *ca(c)*, by *sc(S)*, is equivalent to the invocation of the action on the controlled phenomena via *W*,S^c. This effect upon phenomenon *c* is therefore attempting to set the next state of *W* (a.k.a. ○*W'*)[5]. Ultimately, therefore, our top-level non-operationalisable goal is: avoid harm in the process of achieving $W \Rightarrow\circ W'$. If mixing logics was permissible, but noting a proposition that the left side could be expressed in temporal logic terms, our design goal is to avoid harm in the process of Eq (9). Hence, the discussion under Section 1.3.

$$(W, S_o^c \vdash R) \Rightarrow\circ W' \qquad (9)$$

[4] This appears to be a homage to case-based reasoning graphs, since it separates the rationale graph from the design graph. Or it is attempting to maintain the sacrosanctity of the expression $W, S \vdash R$, since so much is at stake having defined it thus.

[5] In basic temporal logic, a minimal set of operators are: ○=true on next time instance, ◊=eventually true, and □=from now on true. For a pattern library of useful expressions see: https://matthewbdwyer.github.io/psp/

Thus, if goal-oriented approaches have human centricity (Leveson, 2003) then leveraging off goal-oriented concepts from goal-oriented requirements engineering is attractive, especially if we make agency a first-class element. The approach discussed in this section therefore is attractive, if decomposition over any appropriate abstraction hierarchy, with reasoning captured on the fly, can lead to an assurance driven outcome, then that may manage any information lossiness of other approaches (Feodoroff, 2018). Noting, however, Eq (1)..(6) come from a problem-oriented viewpoint, which will not be a problem, as we will discuss below.

1.3 Assurance Cases should have been goal, not claim, oriented all along

The problem for design assurance is the confidence that we can prove, for a system we have designed, that the system by design will so far as is reasonably practicable avoid harm in the process of achieving $W \Rightarrow \circ W'$. Anthropomorphising the problem and borrowing from motivational psychology ideas around goal approaching and anti-goal avoidance, we can recognise that human agents use those two basic techniques pervasively. We can formalise this idea with the notions of liveness and safety properties of a system, as depicted in Table 1, based upon views of multiple authors (van Lamsweerde, 2009) (Moffett, et al., 1996).

Goals	Requirements	Specifications	Property
Achieve	Causes	$C \Rightarrow \Diamond T, C \Rightarrow \circ\, T$	Liveness
Maintain	Sustains	$C \Rightarrow \Box\, T$	
Cease	Terminates	$C \Rightarrow \Diamond \neg T, C \Rightarrow \circ \neg T$	
Avoid	Prevents	$C \Rightarrow \Box \neg T$	Safety

Table 1

The point of Table 1 is twofold. Firstly, it can (left to right) represent the decomposition "stages" between goals, to requirements and down to specifications. In that case we see the differences, left to right, in terms of Leveson's previous explanations of Rasmussen - Explanation 1 and Explanation 2.

Secondly, in Table 1 there is the hint that all three columns could be represented in temporal logic, at distinct levels of refinement. The former idea is, however, the idea we wish to suggest is the design process, which may include a level of semi-formality before ideas are concretized. Modelling formally, top to bottom is expensive. The aim for reasoning semi-formally (argumentatively) being a quality Hall et al. proffer during requirements decomposition because requirements cannot be framed in precise mathematical statements, at least early in the requirements analysis process where natural language may be used (Hall, et al., 2007).

> *... goals are a form of requirement, and goal refinement is a form of requirement transformation that generates a set of sub-goals of a goal.*
>
> Explanation 3

We are encouraged in our approach herein since Hall, Rapanotti and Jackson (Hall, et al., 2007) also invite discussion of goal-orientation with Explanation 3 above.

Essentially therefore, if we assume goals are tuples of ‹*verb,noun*› (verb classes from the "Goals" column of Table 1, or even the "Requirements" column of Table 1), we are simply talking about the requirements engineering act of allocation of a goal to an ‹*agent*› (aka ***User***), or an allocation of a goal to an ‹*actor*› (aka ***Machine***). We therefore have a "requirement" meta-template of:

The ‹*agent/**User***› shall ‹*verb,noun*›, or

The ‹actor/**Machine**› shall ‹verb,noun›

In this case, without inviting a dispute, we take our "players" to either be cognitive agents, or actors (event-based machines). Of course, this may become murky when wanting to argue AI is "cognitive". However, if we understand that BDI agency for example (discussed below under Section 2) is outwardly control-theoretic, then we can take that to be event-based (see Figure 1), with sensed events leading to changes in the environment due to updated actions. After all, we can also take all players, agents included, to be event-based actors through the ideas behind STPA. So, when analysing/designing the deliberational or the computational process of players, our analysis viewpoints should therefore depend upon, as required, the utility that either the agent or the actor aspect of the problem.

The first column "Goals" of Table 1 represents ideas out of motivational psychology that became the core goal intentions in goal-oriented requirements engineering. These goal intentions are interesting since they function as a reasoning modality, of tense, that authors argue is necessary for safety reasoning (Cassano, et al., 2021). Cassano et al. therefore offer a different idea than expressing hazard-oriented concepts translated into a (so-called) claim-oriented language, in notations such as GSN.

The second column "Requirements" of Table 1 comes from a concept for a semi-formal causal logic model for requirements specification (Moffett, et al., 1996). Notice the verbs of the "Requirements" really come from the verb class of the goal intention of the "Goals." It is interesting that this idea of semi-formal causal logic comes from a position that goals are not useful (Jackson, 2000), and yet describes the requirement problem in synonyms of goal intentions.

Similarly, the third "Specification" column of Table 1, using exemplar LTL (Linear Temporal Logic), can represent either the next level of decomposition, or the lowest level of refinement. Temporal logics, of course, have the modality of tense, with actions qualified by "when". Tense is therefore important when modelling causation, and via use of the principles behind control-theoretics, we might also model obligation. We will certainly want to specify the permissible and the impermissible behaviours of a system in our DRDL.

Importantly, Cassano, et al. (Cassano, et al., 2021), draw a comparison of their ideas with that of the goal-oriented requirement engineering approach of van

Lamsweerde[6] (van Lamsweerde, 2009), especially with respect to use of the modality of tense when describing system liveness and safety aspects. Indeed, avoidance (bottom row of Table 1) is used by van Lamsweerde to frame goal expressions of the bypassing or evading aspects of a design, with respect to an unwanted state or condition. Similarly, in STPA, hazards are to be evaded by application of "safety constraints" (Thomas, 2013).

The question begged, therefore: If encoding of hazards was in design terms (or anti-design terms), can we move assurance reasoning into the requirements engineering notations? Certainly, we feel Thomas (Thomas, 2013) expresses hazards in terms of what we might call anti-requirements that we will define as *not-R*, or: $\bar{R}$. We might therefore depict Thomas' hazard of omission as Eq (10) and hazard of commission as Eq (11), and therefore read those concepts into the problem-oriented reasoning style. This idea is captured formally in the AND-OR tables that Thomas describes for documenting both the "R" for requirements and "H" for hazards, albeit without a means to capture the decomposition aspect of the problem.

$$H^{Omission}: W, S_o^{\neg c} \vdash \bar{R} \tag{10}$$

$$H^{Commision}: W, S_o^{c} \vdash \bar{R} \tag{11}$$

With the notion that a hazard may be described as an anti-requirement, then we would fully describe a hazard decomposition as Eq (12), recalling Eq (10) and Eq (11) for the decorations. Of course, what is more natural is to rewrite the anti-requirement in terms of a requirement that avoids the hazardous "zone", thus Eq (13). This is also the idea Thomas (Thomas, 2013) proposes when describing safety constraints (the antithesis of hazards) and then of "responsibilities" which define that liveness path. Each step alone that liveness path aiming to avoid the set of all anti-requirements elicited during STPA denoted as: $\bar{\mathbb{R}}$. The act of taking the antithesis of the hazard, and the setting a safety constraint, helps frame the set of "responsibilities" for all the agents/actors involved in the overall system behaviour (Thomas, 2013). We propose that liveness path description is at Eq (14).

$$\frac{W_1, S_1 \vdash \bar{R}_1; \ldots; W_n, S_n \vdash \bar{R}_n}{W, S \vdash \bar{R}} \tag{12}$$

$$R \notin \bar{\mathbb{R}} \tag{13}$$

$$\frac{W_1, CA_1, S_1 \vdash R_1 \notin \bar{\mathbb{R}}_1; \ldots; W_n, CA_n, S_n \vdash R_n \notin \bar{\mathbb{R}}_n}{W, J, S \vdash R \notin \bar{\mathbb{R}}} \tag{14}$$

One can, therefore, using a semi-formal 2D spatial graphical "causal calculus" depict Eq (14), and thus the notion of liveness, and safety from Table 1, in Figure 1.

This is where we might counter, somewhat, the position of Cassano et al. (Cassano, et al., 2021). Cassano et al. are critical of the ideas van Lamsweerde puts forward but the difference between van Lamsweerde's temporal logic-based decomposition "patterns", and the use of temporal logic-based proofs Cassano et al. recommend, is the difference between a semi-formal means-directed and argumentative approach, and a formal proof-based approach. Given that the proof-based approach aims to prove satisfaction of the ends, then the unanswered question from Cassano et al. is, in any event, from whence do the antecedents of their proofs come? We therefore propose that both the argumentative, especially using means-directed argumentation, and the proof-based approaches are complementary.

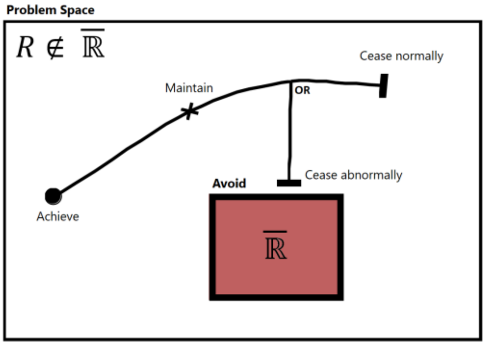

Figure 1

Indeed, safety case reasoners have also adopted ideas from van Lamsweerde, for the (so-called) agent-based pattern-driven decomposition of "arguments" in GSN (Hall-May & Kelly, 2005). In their paper Hall-May & Kelly attempts an agent-based approach using GSN. Hall-May & Kelly borrowing ideas from the notation KAOS (Keep All Objectives Satisfied). Consequently, the approach Hall-May & Kelly adopts does not meet the semantics for safety reasoning that Cassano et al. recommend, as GSN is without either agency or action semantics (Zave & Jackson, 1997; Cassano, et al., 2021).

KAOS is a formal goal-oriented requirement engineering notation that van Lamsweerde has promoted since 1990[6] (van Lamsweerde, 2009). KAOS is, of course, the notational style Cassano et al. seed their position on safety reasoning (Cassano, et al., 2021). Cassano et al. otherwise do not belabour the avoidance style pattern driven approach suggested by van Lamsweerde.

Cassano et al. (Cassano, et al., 2021) themselves are interested in mathematical proofs, and therefore may limit application of their approach to specialists, in an industry that attempts safety in design with generalists. We are not, however, discounting the use of formal proofs, but if we can define a practical reasoning approach that can conceptually span semi-formal to formal thinking then that may facilitate generalists and specialists interoperating meaningfully at any hand off point between those practitioners.

As interesting is that Cassano et al. do not recommend notations such as GSN are useful for safety reasoning (Cassano, et al., 2021). Contrary to Kelly's recommendation not to entertain goal-oriented requirements notations (Kelly, 1998, p. 65), Cassano et al. are looking at goal-oriented requirements engineering for inspiration for a "new" way of safety reasoning for the designer. As the position of Cassano et al. is breaking news, there is no reason not to challenge their objection to goal-oriented requirements engineering approaches. Noting also that we believe that a balance can be struck between the semi-formal and therefore argumentative

[6] https://www.info.ucl.ac.be/~avl/index.php

approaches and also the and formal approaches used in goal-oriented requirements engineering (Yu, 1995) (van Lamsweerde, 2009).

We can see a similar objection towards goal-oriented requirements engineering from GSN practitioners, in the design assurance reasoning space in later works when arguing GSN is a goal-oriented requirement notation (Habli, et al., 2007). It seems; therefore, rationale is required in support of means-directed decomposition, even from proponents of GSN. Despite evidence of objections to "rationale" (Kelly, 1998, p. 65), rationale accounting for the suitability of a means definition, to meet the ends immediately above, exists in the problem-oriented domain as the "Adequacy Argument" of Hall et al. (Hall, et al., 2007) (Hall & Rapanotti, 2009). There are certainly ideas in the goal-oriented requirements engineering domain that conversely challenge the necessity of case-based notations and the thinking behind them (Feodoroff, 2018).

Notable also is that Feodoroff points out that the trend towards counterfactual reasoning in the case-based argument literature, over the last two or more decades, has taken no stock of the fact counterfactual reasoning existed in certain classes of goal-oriented requirements notations (Feodoroff, 2018). That is, counterfactual reasoning for design assurance, in requirement notations, was potentially available already when GSN came to the market (Yu, 1995) (Kelly, 1998). Feodoroff (Feodoroff, 2018) also notes that the counterfactual style reasoning for assurance is to the generalist what the proofs are to the specialists, and that design assurance could therefore be a complimentary or collaborative endeavour, as opposed to a stand-off between factions (argumentation complimented by, rather than versus, proofs). Especially, where a common principle, via decomposition of goal intention, supports the means-directed approach.

Counterfactual reasoning is generally at the core of many belief modification approaches in AI, especially in the world of computational agents using BDI. Noting also that goal satisfaction and the proof of same is already included in certain goal-oriented requirement engineering approaches (Giorgini, et al., 2003). This class of argumentative proof of goal satisfaction is also somewhat akin to the notion of inference networks in AI. Certainly, counterfactual reasoning, being a human-centered process, appears required for our DRDL, if our DRDL is to include argumentation.

2 BDI and Agency

2.1 BDI Roots

In his book, Michael Bratman (Bratman, 1999) discusses his ideas on practical reasoning of humans towards a planning theory of intention. In essence Bratman contends that intention towards goals is facilitated by often prior partial plans as well as the means to assemble those partial plans to enable satisfaction of an overall goal. This essentially therefore requires a means to select partial plans that are goal-directed, to move towards (or away from) a set of states or conditions of the agent.

Agent rationality is a concept Bratman talks about in his book, which is at its core about the deliberation process the agent adopts when selecting partial plans. Bratman, since he is discussing humans, includes problems affecting beliefs and partial plan selection including habits, temperament, competence, trustworthiness, morals, et cetera. Certainly, those aspects are of interest in our analysis of human agents. Belief-Desire-Intention (BDI) is otherwise central also to approaches for a style of computer based computational agency, based upon the ideas Bratman discussed in his book.

Criticism of the BDI style of computational agents is that the three "worlds" of belief, desire, and intention may be limiting according to Rao and Georgeff (Rao & Georgeff, 1995). However, Rao and Georgeff also note that one can roll in additional "worlds" as required. There are a range of other "worlds" that BDI researchers are investigating, with many researchers looking at additional anthropomorphic qualities within the logics of the computational agents.

For the purposes of our discussion herein, however, we will limit our "worlds" to the model for the basic computational agent and that should also marry well with the canonical human agent, but we understand we will need to look for factors that will affect belief formation and plan selection. With belief formation and plan selection being classical sources of accidents where system requirements corrupt those processes. We note also that corrupting of user belief formation and plan selection is otherwise what STPA is attempting to avoid. Codifying hazards in terms of corrupted belief formation or corrupted plan selection, therefore, seems an apt goal for our DRDL.

2.2 BDI and Control Theoretics

In this section we are not trying to supplant current BDI theory, but merely to interpret BDI conceptually against the ideas for assurance driven design practices, especially if that can incorporate control-theoretic thinking. It is as much of interest since STPA, in practice, does not codify goals, beliefs or plans per se – at least as first-class entities. Looking at BDI's mimicking of us, may also potentially be reflected into STPA practices, by attempting to interpret human actors in STPA from our BDI viewpoint herein.

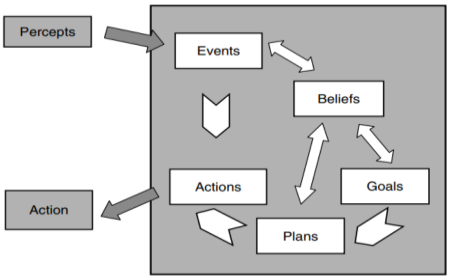

Figure 2

From Figure 2 we note that BDI agents are control-theoretic in the sense of Eq (7), though in the form of Eq (15). That is from Figure 2, we define the object of a perception or Percept p of phenomenon o, and Action a upon phenomenon c. We are, of course, trivialising here because inside the "controller" there could be a range of BDI style logics (Meyer, et al., 2015).

$$S_p^a \tag{15}$$

Continuing our simplification from Figure 2, and reading Eq (15) as $W, S_p^a \vdash R$, we might take Eq (16) for granted using the tuple of ‹*W,E,B,P,G*›; with *p* (percepts) assigned to *E* (events), and *a* (actions) assigned to *P* (plans). Further, taking *a* and *p* for granted, Eq (17) represents a decomposition from *G* to *G*$_1$…*G*$_n$.

$$W, E_p, B, P^a \vdash G \tag{16}$$

$$\frac{W_1, E_1, B_1, P_1 \vdash G_1; \ldots; W_n, E_n, B_n, P_n \vdash G_n}{W, E, B, P \vdash G} \tag{17}$$

So, does Eq (17) explain or justify *G*?

2.3 Explainable versus Justifiable AI

If we take three uses of argument to be rhetoric, explanation, and justification, we can take rhetoric to be this report. In Figure 3, we will take the means-directed decomposition on the left as Explanation, because the assurance industry takes the goal-directed approach on the right to be Justification. Does the means-directed version align, therefore, with explainable AI, and therefore Eq (17)? Certainly, it does according to Harbers et al. (Harbers, et al., 2010).

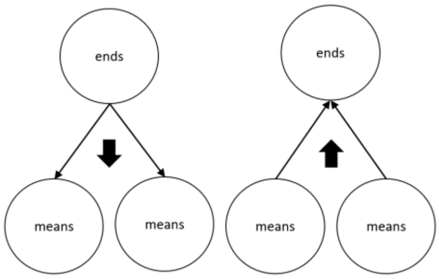

Figure 3

However, does that mean Eq (18) would function as an Explainable or a Justifiable AI approach? Eq (18) is, given its nature, still a means-directed (bottom row "ends" to top row "means") position, thus seems to fit the notion of explanation in style.

$$\frac{W_1, E_1, B_1, CA_1, P_1 \vdash G_1; \ldots; W_n, E_n, B_n, CA_n, P_n \vdash G_n}{W, E, B, J, P \vdash G} \tag{18}$$

As an exercise for the reader, translate Eq (18) into the anti-goal-avoiding equivalent, using the intent of Eq 9. Similarly, translate Eq (10) and Eq (11) into goal-based equivalents for hazards of omission and commission, starting with Eq (16). Once translated, might Eq (19) depict a justifiably safe BDI system? Based upon Eq (14), in Eq (19) includes not-gamma ($\bar{\Gamma}$) as the set of anti-goals (or parts thereof) to avoid (refer again to Figure 1).

$$\frac{W_1, E_1, B_1, CA_1, P_1 \vdash G_1 \notin \bar{\Gamma}_1; \ldots; W_n, E_n, B_n, CA_n, P_n \vdash G_n \notin \bar{\Gamma}_n}{W, E, B, J, P \vdash G \notin \bar{\Gamma}} \tag{19}$$

Essentially, the aim may be to define the system to ensure the means-directed aspects will guarantee the satisfaction of the goal-directed aspects, where the goal is to avoid unwanted system states. Essentially, again, the aim of STPA. We, of course, understand the design time means-directed assurance can still be undone by the real-world.

We need note, however, that Eq (19) may still be read as means-directed, when in a top-down planning phase, and when in a goal-directed execution sense of BDI. However, if the actual plans executed by a computational agent are assemblies of partial plans, and the full set of plans that might be executed are therefore approaching a countable but infinitely large set, and the full set of unwanted anti-goals also cannot be known, then our toy model is not useful, outside our search for semantics for our DRDL, and the design problem.

Whether *J* is necessary is also a question, since *B* is, after all belief, and belief is the basis for justification. Note, however, *J* aims to justify "*the fitness-for-purpose of a found solution*". We believe the *fitness-for-purpose* aspect maps readily to the correlation problem of: *W* versus *B*, or $\rho_{W,B}$. An example of this is the Tesla accident 2016 whereby a Tesla collided with a trailer on an apparently bright day[7], and that Tesla manuals warned that the AI was not guaranteed to identify all hazards. Manuals aside, if the problem of hazard identification was solely in the specification of *W*, then that is simply the Frame Problem[8]. Ultimately, the problem was, more specifically, the likely potential for a low $\rho_{W,B}$ and hence the warning in the manual. Otherwise, on that fateful day we arrive at Eq (20), where $\rho_{W,B}$ was low.

$$\rho_{W,B} \rightsquigarrow G \in \bar{\Gamma} \tag{20}$$

That is, *J* may in this model be in terms, at least initially, of $\rho_{W,B}$. This otherwise seems intuitive given Eq (20) is dealt with in both belief adoption and agent learning research behind computational agents, since the problem is to learn how to avoid the right-hand side of Eq (20).

For our design time problem, and therefore our DRDL, the highest value for an *a priori* $\rho_{W,B}$ is the specification problem. That is, we must aim to somehow set that correlation as high as reasonably practicable, against the best estimate of the extent of the Frame Problem. The rigor suggested by Thomas attempts this goal for the highest value for an *a priori* $\rho_{W,B}$, albeit without a "human" as agent model, recalling Thomas currently aims to describe $\mathbb{R}$ but not $\bar{\Gamma}$ (Thomas, 2013). Otherwise, the actual qualification of $\rho_{W,B}$ will almost always be the *a posteriori* validation of the agent or system success, or failure, in the real world.

That qualification problem will not be solved by a DRDL nor its subsequent graphical notation in and of itself, outside our attempt to support the practical reasoning process, and perhaps assurance, during the design process. However, we can note that any verification process for the system, where we aim to prove that $\rho_{W,B}$ is high, relates to the notion of assurance confidence, especially in the presence of *a posteriori* evidence-based failure-free dependability. That problem of providing evidence-based failure-free dependability correlates as much with the problem for the validation of training sets in machine learning based AI, as it does for non-AI based systems.

Thus, interpreting the problem-oriented model as a meta-model of BDI concepts, at this level of abstraction,

[7] https://www.tesla.com/blog/tragic-loss
https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF

[8] https://plato.stanford.edu/entries/frame-problem/

seems to include useful concepts for our DRDL, especially for human “controllers” in the STPA sense, and because STPA, at least as elaborated by Thomas (Thomas, 2013), attempts the best estimate of $\overline{\mathbb{R}}$ but we could reinterpret the process to also address the best estimate of $\overline{\Gamma}$.

2.4 Belief Adoption versus Assurance Justification.

We suggest that the principal difference between the general problem of belief adoption for agents and assurance Justification is the problem of Trust. The goal of an assurance argument is to provide sufficient probative value to the question of goal satisfaction of important system level goals. That is, assurance Justification traditionally can be defined as the proof of satisfaction of a goal in the *a priori* system (accident) context. We contend that may be generally as difficult a problem as proof of dissatisfaction of an anti-goal, since the aim is to prove (somehow): $\rho_{W,B} \rightsquigarrow G \notin \overline{\Gamma}$.

The aim is thus to argue by attempting to correlate the design context to best fit the operating context of the system. This is obviously a non-trivial task, but it is hoped that supplying practical reasoning tools supports this endeavour. Ultimately trust should, we feel, be a function of comprehensibility of both the design and the argument for the design. Pushing this trust out to the reverse engineering practices of case-based approaches was the problem phased safety cases was trying to remedy (Kelly, 1998), noting therefore that Kelly’s endeavour was similar in intent to Intent Specifications (Leveson, 2003). So, if goal/requirement decompositions can harbour assurance rationale (Hall & Rapanotti, 2009; Feodoroff, 2018), and be of an apt modality (Cassano, et al., 2021) then we feel more effort is required to achieve that end.

3 STPA vs GORE vs ADD vs BDI

Of course, it is a challenging idea that one might attempt a harmonisation across several apparently disparate academic memes, in the absence of a natural gravity such as was present in the case of UML (Unified Modeling Language). However, when taking into consideration the goal and indeed agent orientedness of STPA and problem-oriented Assurance Driven Design (ADD), it does not appear a long reach.

Previously herein we have noted Hall et al. (Hall, et al., 2007) described goals as requirements - Explanation 3. Similarly, we note that proponents of goal-oriented requirements engineering have, when critiquing Eq (1), described requirements as goals (Jureta, et al., 2008). Jureta et al. also see the requirement problem as scale invariant, in the same way as we described above when discussing the approaches of Thomas versus Hollnagel (Section 1.2).

Of note, Jureta at al. (Jureta, et al., 2008) do also introduce a defeasible reasoning style of entailment, using “sceptical consequence”: $A \mid\!\sim B$. Essentially, therefore, Jureta et al. promote the notion of counterfactual and therefore nonmonotonic reasoning, and thus they opt to rewrite Eq (1) as Eq (21). We can therefore agree with Jureta et al., since the symbol $\mid\!\sim$ will mean an argumentative rather than proof-based approach, with commensurate updates of all our other equations herein to replace $\vdash$ with $\mid\!\sim$. That is, use of weighted pros-/cons-based design decisions and the capturing of the rationale for those decisions (Giorgini, et al., 2003; Feodoroff, 2018).

$$W,S \mid\!\sim R \tag{21}$$

The use of sceptical reasoning is certainly part and parcel of the requirement decomposition process when dealing with dead-ends or suboptimal solutions (Hall & Rapanotti, 2009), but is not included in the semantics of the DRDL used by proponents of problem-oriented approaches. This use of sceptical consequence thus falls in line with a latest idea of defeasible assurance reasoning (Rushby, 2021), since the assurance end must be met, but only once all counter arguments are spent[9].

We can note also that Eq (16) is analogous to the goal-based interpretation of Eq (1) by Jureta et al. (Jureta, et al., 2008), though we have in Eq (16) paid attention to the control-theoretics and therefore the operationalisation of functional requirements that Jureta et al. avoided. We feel that Eq (16) better addresses the intent of the approach by Cassano et al. (Cassano, et al., 2021), since Eq (16) literally embodies action and agency. The equivalent $W, S_o^c \vdash R$ to Eq (16) is similarly applicable. Applicable when we note the goal intention behind the rows of Table 1, which will act as the optative expressions of both R and S, are melded with the agency of S and R. This appears especially true when we favour the view that the decorations on the problem-oriented R, perhaps the ⟪*stereotypes*⟫ on R, aim to constrain the solution towards the goal x or, alternatively, constrain the solution away from the anti-goal $\bar{x}$.

The notion of “justified approximation” by Jureta et al. (Jureta, et al., 2008) does not distinguish itself from “Adequacy Argument”, Eq (3), by Hall et al. (Hall, et al., 2007). Hence, goals are requirements are goals Ad infinitum, and the process over the decomposition, it is agreed, is argumentative by problem-oriented, goal-oriented and case-based reasoning niches. The remaining question is therefore around modality, and the modality of tense (aka temporal logic) does marry, as Cassano et al. suggest, with action and agency as safety reasoning (Cassano, et al., 2021). The notion of safety and liveness, and therefore of avoidance style reasoning, which is best expressed as goal intentions using temporal logic - at whatever level of abstraction appears to be apt, in support of goal through requirement through specification decomposition certainly has a place in our DRDL.

Noting that goals are requirements and requirements are goals, we can simply reflect again on Xu et al. (Xu, et al., 2006) and the problem of whether a goal firstly can be operationalized. Once operationalized, by allocation to an actor/agent, thus now in terms of requirement or specification, we are subsequently only considering the

[9] Really the notion of indefeasibility, being a stopping condition for an argument versus any counterargument. Indefeasibility being achieved when all major counter arguments are defeated, and the argument can subsequently deflect trivial counter arguments.

level of decomposition required to sufficiently specify the system.

We note that Eq (21) is also in line with Seator's idea of argument diagrams (Seater, 2009), and the notion of requirement progression, than the original idea of problem progression (Hall, et al., 2005). It is also analogous to ideas behind use of certain classes of goal-oriented requirements notations for assurance reasoning during requirement engineering (Feodoroff, 2018), since those notations graphically allocate goals and tasks to actors/agents, along with both monotonic and nonmonotonic reasoning semantics (Giorgini, et al., 2003).

There is an interesting sidebar to the story of control-theoretics and the problem-oriented approach that requires our attention. It essentially touches on the patterns used in the graphical aspect of the problem-oriented approach. The traditional graphical patterns exclude, by design, the idea of control-theoretics (Jackson, 2000)[10]. Jackson does this by noting the 4-variable model by Parnas and Maddy (Parnas & Madey, October 1995), but describes the 4-variable model as not useful for the general problem of software specification. The interpretation of the 4-variable model by Jackson does not, however, threaten the sacrosanctity of Eq (1).

The 6-variable model of Ulfat-Bunyadi et al. (Ulfat-Bunydai, et al., 2016), however, does upset the sacrosanctity of Eq (1). The 6-variable model aims to inject a ***System***, between the ***User*** and the ***Machine*** in the problem-oriented model. The 6-variable model therefore uses two layers of abstraction as per Eq (22). The two layers of abstraction are used to systematically decompose a user requirement REQ_{RW}, a system requirement REQ, and then REQ into a software requirement SOF. Further, the world NAT_{RW} is decomposed to domains of phenomenon NAT. This is literally a top-down decomposition, with a means-directed step of REQ and because NAT_{RW} decomposes to NAT in the same way that W decomposes to W_n in Eq (2). Each of the two layers can also be decomposed as per Eq (2), for essentially decomposition with the abstraction layers, importantly, when control-theoretic is the consideration.

$$\begin{aligned} NAT_{RW}, REQ &\vdash REQ_{RW} \\ NAT, SOF &\vdash REQ \end{aligned} \tag{22}$$

The idea of layers of abstraction of decomposition was previously described by van Lamsweerde (van Lamsweerde, 2009), when van Lamsweerde interprets problem-oriented thinking. Without complicating the equations, therefore, one can build a hierarchy of abstraction as expressions per Eq (23). Thus, we start with P for Problem solved, decomposed to G for Goals, decomposed to R for Requirements, and thence onto S for Specifications. Noting the world domain W is constrained at each layer by the domain phenomenon expressed at the same level of abstraction of the "end", or *succedent*, on the right of the expression ($\lceil P \rceil$, $\lceil G \rceil$ or $\lceil R \rceil$), Vertically, the domains would be related by Eq (24).

We, of course, can rewrite the individual layers of Eq (23), because of Eq (7) and Eq (15), as either of Eq (14) or Eq (19). Not forgetting the replacement of formal entailment with the "sceptical consequence" version of $\mathrel{|\!\sim}$ to facility the collection of nonmonotonic reasoning over the graph.

Ultimately, Eq (23) takes us back somewhat to hierarchical task analysis, and just the problem of systematically breaking down the system. This model thus aligns with the goal-oriented decomposition requirements of an Intent Specification concept, which Leveson claims are a means of creating human-centred specifications (Leveson, 2003).

$$\begin{aligned} W_{\lceil P \rceil}, G &\vdash P \\ W_{\lceil G \rceil}, R &\vdash G \\ W_{\lceil R \rceil}, S &\vdash R \end{aligned} \tag{23}$$

$$\frac{W_{\lceil G \rceil_1}, \dots, W_{\lceil G \rceil_n}}{W_{\lceil P \rceil}}, \frac{W_{\lceil R \rceil_1}, \dots, W_{\lceil R \rceil_n}}{W_{\lceil G \rceil}} \tag{24}$$

Work to go includes developing a DRDL, based upon the concepts herein, prior to moving onto a design of a graphical notation.

4 Conclusion

The fact that Cassano et al. are, in no small measure, asking such a confronting question as "what is safety reasoning" in 2021, should of course be tackled (Cassano, et al., 2021). It seems if actions, modality and agency are necessary for safety reasoning, that certainly falls into the domain of goal-oriented requirements engineering, especially where agent formalisms are included (Zave & Jackson, 1997).

If sound and complete theories for realistic systems are elusive (Hall, et al., 2005) (Jureta, et al., 2008), at least in the preliminary stages of a system specification, then a top-down argumentative process during requirement specification seems apt. If that is framed in theoretics that can provide means-directed assurance driven rationale capture, that also seems apt. Rationale being what Kelly described as "aspirational", a.k.a. the intentional goal form (‹*achieve,noun*›), versus the past participle form of the so-called claim (‹*achieved,noun*›) (Kelly, 1998, p. 65). The claims still to be justified by the presence of evidence that the goals are satisfied, and the anti-goals are dissatisfied, with sufficient evidence-based failure-free dependability.

If safety reasoning is therefore more a case of empathy for the user in terms of goal satisfaction, or of anti-goal dissatisfaction, then paying attention to both the deliberational and the computational aspects of players within the design is necessary. Given the modality of tense for modelling causation, with the control-theoretics used to describe the permissible safe behaviour and therefore the deontic modality, it must all fall therefore onto the documenting of the practical rationality of the design, during the design process, to provide assurance confidence via expression of agency and action within the design.

[10] A curious omission, since we have also noted the control-theoretics of the phenomenon decoration pair of o and c, and thus S_o^c.

5 Bibliography

Abdalla, R. & Mishra, A., 2021. Agent-Oriented Software Engineering Methodologies: Analysis and Future Directions. *Complexity,* December.2021(1629419).

Bratman, M. E., 1999. *Intention, Plans and Practical Reason.* 1st ed. Stanford, California, USA: CSLI Publications.

Cassano, V., Maibaum, T. S. E. & Grigorova, S., 2021. Towards Making Safety Case ArgumentsExplicit, Precise, and Well Founded. In: *Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems.* Singapore: Springer, pp. 227-258.

Feodoroff, R., 2018. *Back to the Future: Pollock versus Toulmin.* Melbourne, Victoria, Australia, https://ascsa.org.au/conferences/2018/#technical-program.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E. & Sebastini, R., 2003. Formal Reasoning Techniques for Goal Models. *Journal on Data Semantics,* Volume 2800, pp. 1-20.

Habli, I., Wu, W., Attwood, K. & Kelly, T., 2007. Extending Argumentation to Goal-Oriented Requirements Engineering. *Lecture Notes in Computer Science,* Volume 4802, pp. 306-316.

Hall, J. G. & Rapanotti, L., 2009. Assurance-driven design in Problem Oriented Engineering. *International Journal On Advances in Systems and Measurements,* 2(1), pp. 119-130.

Hall, J. G., Rapanotti, L. & Jackson, M., 2005. Problem frame semantics for software development. *Software and Systems Modeling,* 4(2), pp. 189-198.

Hall, J., Mannering, D. & Rapanotti, L., 2007. *Arguning Safety with Problem Oriented Software Engineering,* Milton Keynes, Buckinghamshire, England: Department of Computing, Faculty of Mathematics and Computing, The Open University.

Hall-May, M. & Kelly, T., 2005. Defining and Decomposing Safety Policy for Systems of Systems. *Lecture Notes in Computer Science,* Volume 3688, pp. 37-51.

Harbers, M., van den Bosch, K. & Meyer, J.-J., 2010. *Design and Evaluation of Explainable BDI Agents.* Toronto, ON, Canada, 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

Hollnagel, E., 2012. *FRAM – The Functional Resonance Analysis Method.* 1st ed. Farnham, UK: Ashgate.

Jackson, M., 2000. Problem Frames: Analysing and Structuring Software Development Problems. 1st ed. Lebanon, Indiana, USA: Addison-Wesley.

Jureta, I., Mylopoulos, J. & Faulkner, S., 2008. *Revisiting the Core Ontology and Problem in Requirements Engineering.* Barcelona, Spain, 16th IEEE International Requirements Engineering Conference, pp 71-80.

Kelly, T., 1998. *Arguing Safety: A Systematic Approach to Managing Safety Cases,* York, England: Department of Computer Science, University of York.

Leveson, N., 2003. *Intent Specifications: An Approach to Building Human-Centered Specifications.* [Online] Available at: http://www.safeware-eng.com/system%20and%20software%20safety%20publications/Intent%20Specifications.htm
[Accessed 12 August 2023].

Leveson, N., 2017. Rasmussen't Legacy. *Applied Ergonomics,* 59(March), pp. 581-591.

Moffett, J., Hall, J., Coombes, A. & Mcdermid, J., 1996. A model for a causal logic for requirements engineering. *Requirements Engineering,* 1(1), pp. 27-46.

Parnas, D. L. & Madey, J., October 1995. Functional documents for computer systems. *Science of Computer Programming ,* 25(1), pp. 41-61.

Rao, A. S. & Georgeff, M. P., 1995. *BDI Agents: From Theory To Practice.* San Francisco, California, USA, https://aaai.org/proceeding/01-icmas-95/.

Rasmussen, J., Pejtersen, A. M. & Schmidt, K., 1990. *Taxonomy for Cognitive Work Analysis (Risø-M-2871),* DK-4000 Roskilde, Denmark: Risø National Laboratory.

Rushby, J., 2021. The Indefeasibility Criterion for Assurance Cases. In: Y. A. Ameur, S. Nakajima & D. Mery, eds. Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems, Communications of NII Shonan Meetings. Kamiyamaguchi, Hayama Miura-gun, Kanagawa, Japan: Springer, pp. 259-279.

Seater, R. M., 2009. *Building Dependability Arguments for Software Intensive Systems,* Cambridge, Massachusetts, USA : MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

Thomas, J., 2013. EXTENDING AND AUTOMATING A SYSTEMS-THEORETIC HAZARD ANALYSIS FOR REQUIREMENTS GENERATION AND ANALYSIS, Cambridge, Massachusetts, USA : MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

Ulfat-Bunydai, N., Meis, R. & Heisel, M., 2016. The Six-Variable Model - Context Modelling Enabling Systematic Reuse of Control Software. *Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016) - ICSOFT-PT,* 0ICSOFT(1), pp. 15-26.

van Lamsweerde, A., 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications.* 1st ed. Chichester, West Sussex, England: John Wiley & Sons.

Xu, L., Ziv, H., Alspaugh, T. A. & Richardson, D. L., 2006. An architectural pattern for non-functional dependability requirements. *Journal of Systems and Software,* 79(10), pp. 1370-1378.

Yu, E. S. K., 1995. *Modelling Strategic Relationships For Process Reengineering,* Toronto, ONT, Canada: Dept. of Computer Science, University of Toronto .

Zave, P. & Jackson, M., 1997. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology,* 6(1), pp. 1-30.