Technical Report: A Refinement Calculus for Requirement Engineering based on Argumentation Theory

Alex Borgida[1]*, Yehia Elrakaiby[2], Alessio Ferrari[3], and John Mylopoulos[4]

[1] Rutgers University, New Brunswick NJ, USA borgida@rutgers.edu
[2] Université du Luxembourg, Luxembourg, yehia.elrakaiby@uni.lu
[3] CNR-ISTI, Pisa, Italy alessio.ferrari@isti.cnr.it
[4] University of Toronto and University of Trento, jm@cs.toronto.edu

Abstract. The Requirements Engineering (RE) process starts with initial requirements elicited from stakeholders – however conflicting, unattainable, incomplete and ambiguous – and iteratively refines them into a specification that is consistent, complete, valid and unambiguous. That specification consists of functions, quality constraints and assumptions on the environment of the system-to-be. We propose a novel RE process in the form of a calculus called CaRE (Calculus for Requirements Engineering) where the process is envisioned as an iterative application of refinement operators, with each operator removing a defect from the current requirements. Our proposal is motivated by the dialectic and incremental nature of RE activities. The calculus casts the RE problem as an iterative argument between stakeholders, who point out defects (ambiguity, incompleteness, etc.) of existing requirements, and then propose refinements to address those defects, thus leading to the construction of a refinement graph. This graph is then a conceptual model of an RE process enactment. The semantics of these models is provided by Argumentation Theory, where a requirement may be attacked for having a defect, which in turn may be eliminated by a refinement.
The detailed proposal of CaRE has been submitted for consideration to the 39th International Conference on Conceptual Modeling (ER 2020). In this technical report, we illustrate an application scenario for CaRE to showcase the proposed calculus and approach.

Keywords: requirements engineering, RE process, argumentation theory

1 Scenario: ERTMS/ETCS Level 3

We showcase the proposed calculus on an example of a train control system, inspired from the ERTMS/ETCS Level 3 moving block system [2, 1]. The ERTMS/ETCS system is the European standard for train control and management. In its deployed Level 2 implementations, the location of trains is identified by means of fixed devices, called *balises*, which are positioned along the track. The balise communicates its location when the train passes over it, so that the train has an accurate, yet *intermittent*, information about its position, which is shared with the central control system. In this way, the

[*] Authors in alphabetical order.

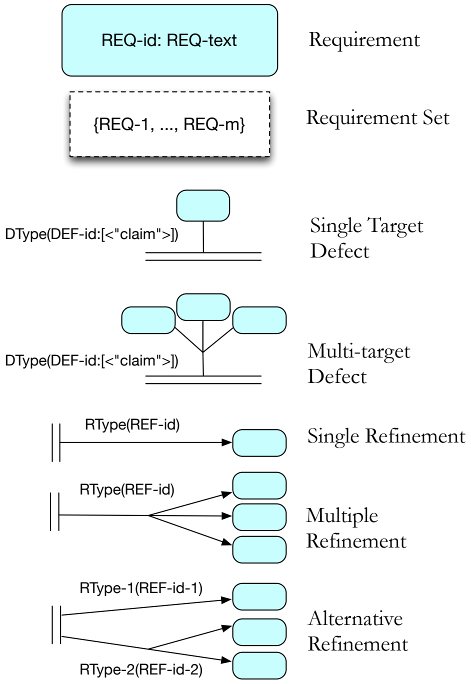

Fig. 1. Graphical Notation

central system can provide train separation by allocating fixed segments of line, called *block sections*, to each single train. The length of the segment is established based on the distance between balises. With the ERTMS/ETCS Level 3, currently under study, the location of trains is identified by means of satellite navigation, which enables *continuous* train positioning. If the central system can identify the exact location of each train, more trains can be safely routed along the same track. This concept is known as *moving block*, since the block section, i.e., the segment of line allocated to a train, can vary in a continuous and dynamic way, depending on the position of the preceding train.

2 Application of CaRE

In our scenario, we consider a fictional negotiation process between the stakeholders involved in the definition of the requirements of ERTMS/ETCS Level 3, showing how our approach could be applied. Although based on a simplification of the problem, we believe that the example is representative of issues that may emerge in a real-world requirements negotiation scenario. We consider only two subjects involved in the negotiation, the stakeholder (S), representative of the different parties who have an interest in the development of the system, and the requirements analyst (A), representative of the party who will develop the system.

The basic graphical elements of the refinement graph are in Fig. 1. Fig. 2 reports the refinement graph for the scenario. Numbers in boxes associated to requirements—not currently part of the notation—identify the step in which a certain refinement took place.

Step 0: The initial requirements proposed by S are:

- g00: The system shall ensure safety distance between trains
- g01: The distance between trains shall be minimal
- g02: The train location shall be identified through satellite navigation

Step 1: A argues that the provided requirements are not associated to justifications (unjustified defects d01, d02, d03), and S provides them for each requirement (justify refinements r10, r11, r12).

- g00 → g10: avoid train collisions
- g01 → g11: maximise line capacity
- g02 → g12: reduce deployment costs
- g02 → g13: reduce maintenance costs
- g02 → g11: maximise line capacity

Step 2: g00 is too abstract (nonAtomic(d03)), and S performs a first attempt to decompose it to come towards more refined requirements (reduce(r20)).

- g00 → g20: the system shall be composed of a wayside system and an onboard system
- g00 → g21: the wayside system shall indicate to the onboard system the distance that the train can safely travel (Movement Authority, MA)
- g00 → g22: the onboard system shall ensure that the train is stopped at the end of the MA

Step 3: g21 and g22 are still too abstract, and further decomposition is performed.

- g21 → g30: the wayside system shall identify the location of each train on the track
- g21 → g31: the wayside system shall compute the MA for each train on the track
- g22 → g32: the onboard system shall identify its location
- g22 → g33: the onboard system shall compute the breaking curve to ensure that the train is stopped at the end of the MA

Step 4: Further refinements are as follows:

- g30 → g40 the onboard system shall send the location of the train to the wayside system
- g30 → g41 the wayside system shall receive the location of each train on the track

Step 5: A argues that the term “minimal”, used in g01, is vague (ambiguous(d04)), so S clarifies it (clarify(r50)).

- g01 → g50: the distance between trains is the distance between the front of a train and the rear of the preceding train
- g01 → g51: the breaking distance is the distance that the train needs to travel to come to a stop from its current speed
- g01 → g52: the distance between trains is minimal if it is equal to the breaking distance

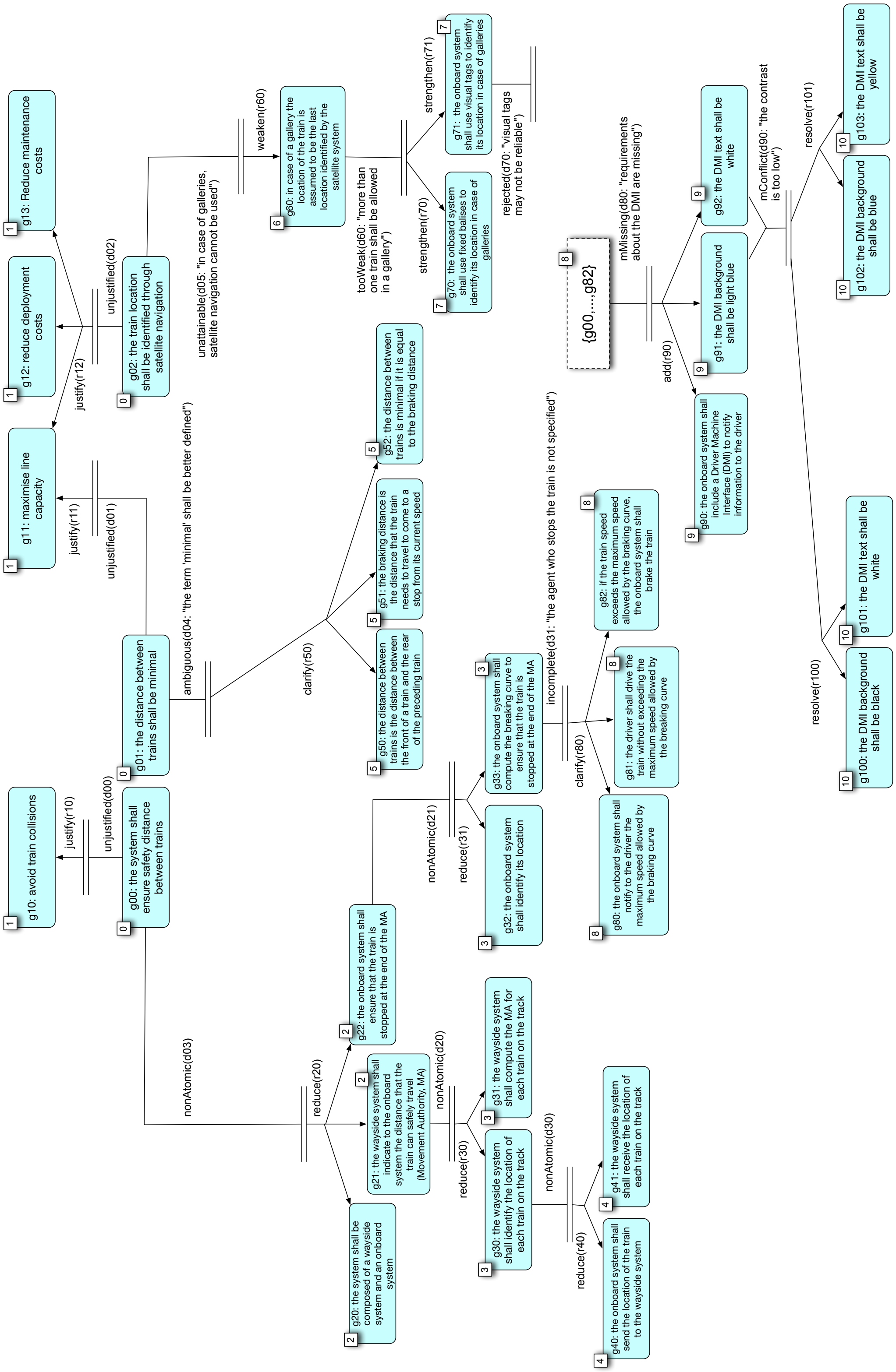
1 g10: avoid train collisions
justify(r10)
unjustified(d00)
0 g00: the system shall ensure safety distance between trains
nonAtomic(d03)
reduce(r20)
2 g20: the system shall be composed of a wayside system and an onboard system
2 g21: the wayside system shall indicate to the onboard system the distance that the train can safely travel (Movement Authority, MA)
2 g22: the onboard system shall ensure that the train is stopped at the end of the MA
nonAtomic(d20)
reduce(r30)
3 g30: the wayside system shall identify the location of each train on the track
3 g31: the wayside system shall compute the MA for each train on the track
nonAtomic(d30)
reduce(r40)
4 g40: the onboard system shall send the location of the train to the wayside system
4 g41: the wayside system shall receive the location of each train on the track
nonAtomic(d21)
reduce(r31)
3 g32: the onboard system shall identify its location
3 g33: the onboard system shall compute the breaking curve to ensure that the train is stopped at the end of the MA
incomplete(d31: "the agent who stops the train is not specified")
clarify(r80)
8 g80: the onboard system shall notify to the driver the maximum speed allowed by the braking curve
8 g81: the driver shall drive the train without exceeding the maximum speed allowed by the breaking curve
8 g82: if the train speed exceeds the maximum speed allowed by the braking curve, the onboard system shall brake the train
1 g11: maximise line capacity
justify(r11)
unjustified(d01)
0 g01: the distance between trains shall be minimal
ambiguous(d04: "the term 'minimal' shall be better defined")
clarify(r50)
5 g50: the distance between trains is the distance between the front of a train and the rear of the preceding train
5 g51: the braking distance is the distance that the train needs to travel to come to a stop from its current speed
5 g52: the distance between trains is minimal if it is equal to the braking distance
1 g12: reduce deployment costs
1 g13: Reduce maintenance costs
justify(r12)
unjustified(d02)
0 g02: the train location shall be identified through satellite navigation
unattainable(d05: "in case of galleries, satellite navigation cannot be used")
weaken(r60)
6 g60: in case of a gallery the location of the train is assumed to be the last location identified by the satellite system
tooWeak(d60: "more than one train shall be allowed in a gallery")
strengthen(r70)
strengthen(r71)
7 g70: the onboard system shall use fixed balises to identify its location in case of galleries
7 g71: the onboard system shall use visual tags to identify its location in case of galleries
rejected(d70: "visual tags may not be reliable")
8 {g00,...,g82}
mMissing(d80: "requirements about the DMI are missing")
add(r90)
9 g90: the onboard system shall include a Driver Machine Interface (DMI) to notify information to the driver
9 g91: the DMI background shall be light blue
9 g92: the DMI text shall be white
mConflict(d90: "the contrast is too low")
resolve(r100)
resolve(r101)
10 g100: the DMI background shall be black
10 g101: the DMI text shall be white
10 g102: the DMI background shall be blue
10 g103: the DMI text shall be yellow

Fig. 2. Refinement Graph for the ERTMS/ETCS Level 3 application scenario.

Step 6: A argues that satellite navigation required by g02 may not work in case of galleries, and this makes the requirement unattainable. A specific requirement for galleries is defined that weakens the original requirement (weaken(r60)).

- g02 → g60: in case of a gallery the location of the train is assumed to be the last location identified by the satellite system

Step 7: g60 is not acceptable for S, since it implies that only one train is allowed to move in a gallery (the requirement is considered tooWeak). To strengthen this requirement, A proposes to either use the traditional fixed balise system to locate the train in galleries (strengthen(r70)), or to use visual tags placed along the gallery (strengthen(r71)), which should be detected by a camera on the train. This second option is rejected, as this system may not be sufficiently reliable.

- g60 → g70: the onboard system shall use fixed balises to identify its location in case of galleries
- g60→ g71: the onboard system shall use visual tags to identify its location in case of galleries

Step 8: By reviewing the requirements, A notices that g33 does not specify *who* is stopping the train. The requirement is deemed incomplete (incomplete(d31)). S specifies that the driver is in charge of driving and stopping the train, while the onboard system monitors that the breaking curve is respected. If the driver violates the braking curve, the onboard system brakes the train. Hence, the requirement is clarified as follows:

- g33 → g80: the onboard system shall notify to the driver the maximum speed as allowed by the braking curve to stop at the end of the MA
- g33 → g81: the driver shall drive the train without exceeding the maximum speed allowed by the braking curve
- g33 → g82: if the train speed exceeds the maximum speed allowed by the braking curve, the onboard system shall brake the train

Step 9: A understands that there is a human agent in the system (the train driver), but no requirement is specified concerning the communication with this agent. The requirements set is considered incomplete (mMissing(d80)). Therefore, A solicits S to discuss driver-related requirements. The following requirements are introduced (add(r90)):

- g90: the onboard system shall include a Driver Machine Interface (DMI) to notify information to the driver
- g91: the DMI background shall be light blue
- g92: the DMI text shall be white

Step 10: A observes that the contrast may be insufficient if the background is light blue and the text is white (mConflict(d90)). Two options are proposed to resolve the conflict.

- g91, g92 → g100, g101: the DMI background shall be black, the DMI text shall be white
- g91, g92 → g102, g103: the DMI brackgroun shall be blue, the DMI text shall be yellow

We run the tool on the previous example using a MacBook pro, with a 2.2 GHz Intel i7 processor and 32 GB of DDR RAM. The tool read the refinement graph and identified its initial and specification nodes in approximately 23ms. The computation of the minimal specifications took around 280ms producing the two specifications: {g20, g31, g32, g40, g41, g50, g51, g52, g70, g80, g81, g82, g90, g100, g101}, and {g20, g31, g32, g40, g41, g50, g51, g52, g70, g80, g81, g82, g90, g102, g103}.

This scenario can be run using the tool that is made available at the following DOI: 10.5281/zenodo.3856402.

References

1. Basile, D., ter Beek, M.H., Ferrari, A., Legay, A.: Modelling and analysing ERTMS L3 moving block railway signalling with simulink and uppaal SMC. In: Larsen, K.G., Willemse, T.A.C. (eds.) Formal Methods for Industrial Critical Systems - 24th International Conference, FMICS 2019, Amsterdam, The Netherlands, August 30-31, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11687, pp. 1–21. Springer (2019). https://doi.org/10.1007/978-3-030-27008-7_1
2. European Union Agency for Railways: ERTMS/ETCS System Requirements Specification 3.4.0. http://www.era.europa.eu/Document-Register/Pages/Set-2-System-Requirements-Specification.aspx (2016)