

Exploring Goal Relationships in Satellite Assembly Line Design

ANOUCK CHAN & THOMAS POLACSEK & STÉPHANIE ROUSSEL*

Abstract

With the emergence of satellite constellations in recent years, satellite assembly lines have to respond to new demand : produce multiple copies of the same satellite at short production rates, manage multiple production rates for different constellations, but also continue to produce specific satellites in a single copy over longer time horizons. Designing versatile assembly lines that can assemble multiple types of satellites is a major challenge for manufacturers. Such lines must be designed within the existing infrastructure and with the specific required machines and tools, which must be shared between assembly activities for all types of satellites. In this work, we are interested in eliciting the goals of a versatile assembly chain from the perspective of goal-requirement modelling. More specifically, we want to determine which goals might conflict with each other. To do this, we propose to elicit the different goals and then use a constraint programming approach to compute if some goals are difficult to satisfy together.

1. INTRODUCTION

Satellites are complex systems composed of various components such as solar panels, communication systems, propulsion systems and scientific instruments. While satellite production historically involved low-volume, the recent demand for satellite constellations has necessitated a shift towards high-volume manufacturing. This new demand creates a need for the exploration of alternative production methods. A satellite constellation is a group of satellites working together in a coordinated manner to achieve a specific purpose. These constellations offer a wide range of functionalities including observation, communication or scientific research. *Communication constellations* enable high-speed internet access across the globe and are composed of thousands of small satellites in low Earth orbit, *global navigation satellite systems* provide precise positioning and timing information for navigation and various location-based services, *Earth observation constellations* are equipped with sensors to observe and monitor the Earth's surface, atmosphere and oceans. They provide insights for environmental monitoring, disaster management and resource management. Examples include the *Copernicus Sentinel* constellation and the *Dove* constellation of small satellites, showcasing the diverse applications of these coordinated satellite systems.

Despite potential variations in the specific requirements of different satellite types, their similarities in assembly processes and components make them suitable for shared production lines. This shared approach is facilitated by the common aspects of both the assembly procedures and the components in the different satellite categories. Many essential components, such as solar panels, attitude control systems, and propulsion systems, are common in different satellite types. In addition, the assembly processes also show a high degree of similarity for diverse satellite types. Processes can include tasks such as structural component integration, payload instrument or communication payload installation, and rigorous testing and verification procedures. Consequently, the assembly line can be designed to handle the assembly of these common components reducing the need for specialised equipment and processes.

*Authors version, Exploring Goal Relationships in Satellite Assembly Line Design

While the fundamental principles of assembly line design can be applied to satellite production, the unique requirements and complexities of satellite manufacturing necessitate a nuanced and specialised approach. This necessitates expertise and careful consideration of factors like customisation and technological complexity. Therefore, constructing different satellite types on a singular assembly line presents a significant challenge that can provide economies of scale and efficiency benefits. By maximising the utilisation of manufacturing equipment and resources, manufacturers can achieve cost reductions and faster production cycles. However, designing such an assembly line requires the understanding of the trade-offs involved. Should the line be flexible and adaptable to accommodate technological advancements, evolving customer requirements, and fluctuating market trends? Should it be easy to scale the production up or down in response to market dynamics? Should the focus be on minimising capital costs by sharing resources across multiple satellite constellation projects, or reducing operating costs? Defining clear and prioritised objectives is crucial before designing such an assembly line. An optimised line requires a clear understanding of the criteria being optimised, as well as the identification of potentially conflicting objectives that necessitate strategic trade-offs. Furthermore, setting clear goals and requirements is essential for the design of a versatile satellite assembly line. By defining specific objectives and criteria upfront, manufacturers can ensure that the line is designed to effectively address the key challenges and opportunities inherent in the satellite manufacturing process.

Goal-Oriented Requirements Engineering (GORE) is an approach that focuses on capturing and defining the objectives, the goals, of stakeholders to drive the development of systems [15]. Unlike requirements, goals are not necessarily mandatory but rather desires, desirable objectives. It is therefore possible for goals to be in opposition. One major contribution of GORE is to finely characterize the interactions between goals. Frameworks such as *NFR (Non-Functional Requirements)* [14, 5], *iStar* [19, 6] and *GRL (Goal-oriented Requirement Language)* [1] proposed to model these links between goals. These relationships help to define the dependencies, conflicts and synergies between different goals within the system being modelled. Understanding these relationships is critical to ensure that the requirements derived from the goals are coherent and consistent. Usually, goal relationship elicitation involves collaborative efforts among stakeholders and requirements engineers. In the context of a project to investigate the versatile satellite assembly line feasibility, we have used a goal-based approach. Goal elicitation has enabled us to define what a *good* versatile assembly line could be. In order to be able to choose between these goals, it is also necessary to determine which goals reinforce each other and which goals are difficult to combine. However, defining such relations in our context was far from trivial. For example, given the complexity of the interactions involved in satellite manufacturing, determining if the satisfaction of the objective *keeping to a production schedule* has an impact on the satisfaction of *need to avoid storage space* is not simply a matter of expert judgement. Therefore, in order to find non-trivial relationships between goals, we have used automatic computation. As we shall see, we have not carried out a systematic and exhaustive study, but we have laid the foundations of an approach that could enrich traditional GORE approaches. On the basis of real industrial data, using our approach based on automatic computation, we have been able to find conflict between goals that were not elicited by the experts.

The paper is structured as follows: in Section 2, we present the challenges of designing a satellite assembly line and how we have encountered the problem of defining what constitutes good design. We will then explore how goal modeling can assist us, in Section 3, and present a goal model for a versatile satellite assembly line in Section 4. Section 5 is dedicated to our proposal to automatically calculate possible interactions between goals. Section 6 presents related work on the integration of calculus and goal models. Finally, we conclude in Section 7 with a brief discussion on the limitations of the methods and future work.

2. DESIGN A GOOD ASSEMBLY LINE

In our context, an assembly line is a set of physical work stations, composed of jigs and fixtures. Each type of satellite requires a specific set of operations to be performed. These operations require the use of dedicated jigs, fixtures and tools. Satellites under construction are moved from station to station according to the operation that needs to be performed.

Our use case aims to create a versatile manufacturing facility capable of assembling a variety of satellites, including two satellite constellations and two specific earth observation satellites. The assembly line should be designed with modularity in mind, allowing seamless integration of different satellite components and subsystems. It should allow multiple satellites to be processed in parallel, optimising production efficiency and reducing lead times. In addition, flexible workflow configurations could be a key issue, allowing easy reconfiguration of assembly processes to accommodate different satellite designs and production schedules. Moreover, cost reduction through the sharing of jigs, test facilities, and specific tooling is a strategic approach that can significantly reduce overheads and improve efficiency.

Integration jigs and fixtures are used to hold components in place during assembly and integration. Jigs and fixtures are often custom-designed to accommodate the specific geometry and requirements of each satellite type. Satellite components are delicate and often need to be handled with care. Equipment such as robotic arms, lifting devices and handling carts are used to safely transport components within the cleanroom. A cleanroom is a controlled environment in which the concentration of airborne particles such as dust, microbes, aerosol particles and chemical vapours is kept within specified limits. In practical terms, a cleanroom is an area of the production line equipped with high-efficiency particulate air (HEPA) filters to remove airborne particles. The airflow within the cleanroom is carefully controlled to ensure that particles are continuously removed from the environment and that contaminants are not introduced into sensitive areas.

In assembly, alignment and measurement tools play a critical role in ensuring that components are accurately positioned and aligned according to design specifications. For example, interferometers are used to measure distances and surface flatness with high precision using interference patterns created by beams of light. For very sensitive tasks, vibration isolation systems could be used. These features are designed to minimise external disturbances that could affect the precision and quality of the assembly processes.

In addition, throughout the assembly process, a range of test equipment is used to ensure that each component works correctly. This equipment can be very expensive. For example, blackout enclosures are specially designed chambers with opaque walls to block external light sources. These enclosures create a dark environment suitable for carrying out optical tests without interference from ambient light. In fact, satellite optical devices such as imaging sensors, cameras and telescopes require precise measurements and evaluations that can be affected by ambient light.

In practice, as part of our case study, we are aiming to design an assembly line capable of manufacturing three different types of satellite (we are working on a real industrial delivery schedule). Type 1 and Type 2 satellites are part of the constellation and have to be delivered at regular intervals, on specific dates. Type 1 satellites must be delivered every two months and Type 2 satellites every three months. In the end, on a two years period, four Type 1 and five Type 2 satellites must be produced. Type 3 satellites are *one-off* satellites, they do not have to be delivered at regular intervals but have a final delivery date. The factory where the assembly line will be installed already has tools and machines. So, in the first stage of our work we want to study the feasibility of a versatile plant based on the resources available. Minimising and maximising performance criteria or resource consumption are not part of our objectives.

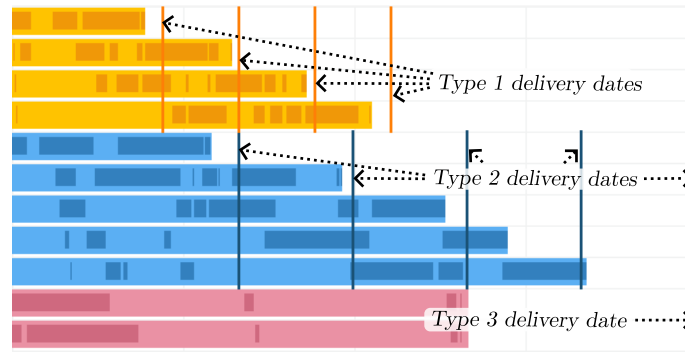


Figure 1: Case Study, Assembly Line Scheduling

We have encoded our problem as a constraint optimisation problem (using CpOptimizer 20.1 [11] and the open-source solver ACE [12]). For this first application, we have no objective function (*i.e.* there is no criterion to optimise), and only criteria to satisfy. We obtain the assembly line whose schedule is given in Figure 1. If the computer finds no solution, it means that building an assembly line is impossible. In the figure, each line represents the schedule for assembling a satellite. The colours yellow, blue and pink represent Type 1, 2 and 3 satellites respectively. The yellow, blue and pink vertical bars represent the dates by which a Type 1, 2 or 3 satellite must be delivered. So we see that it is possible to build a versatile factory, but is it a good factory? How to define *good*?

3. GOAL MODELING

To define *good*, we can focus on the storage. In our solution, presented Figure 1, satellites are manufactured well in advance of their delivery dates. Perhaps it is better not to store them and to assemble them as late as possible? Secondly, we only considered with the existing tools and facilities. It may be possible to go faster with additional equipment, such as HEPA or interferometers. In fact this can be useful to reduce bottlenecks, but it increases the investment. It may also be interesting to minimise the amount of equipment used. This equipment could be used on another assembly line, thus reducing investment costs. These are just a few examples of the goals that could define a good versatile satellite assembly line.

In order to define what is a good assembly line, we opted for goal modelling, drawing upon the principles of *Goal-Oriented Requirements Engineering (GORE)*. This selection was motivated by several factors.

The goal modelling framework helps us to organise these elements by establishing relationships and guiding the elicitation of intermediate goals through *why* and *how* questions [18]. Then significant portion of the acquired goal has fuzzy description and ambiguity in determining their satisfaction. Because they are vaguely defined goals with no clear-cut criteria for their fulfillment, soft goals are particularly well suited to modelling this type of objective [7]. A third reason is that conflicts may arise amongst the elicited goals, where satisfying one might hinder the fulfillment of another. Additionally, redundancy may emerge due to differing elicitation methods employed by experts or the need for refined formulations and additional precision. Therefore, a flexible framework that accommodates redundancy and conflict expression was crucial.

Given these three main reasons, coupled with prior positive experiences in modelling, we choose goal modelling diagram, specifically the *Goal-oriented Requirements Language (GRL)* frame-

work, as the foundation to design an versatile assembly line performance model [1].

4. GOALS FOR A GOOD ASSEMBLY LINE

In the context of our research, it is crucial to differentiate between a mere functional assembly line design and a good one. To achieve this distinction, we must identify a set of evaluation elements, more specifically, goals. Our goal model was constructed through a two-pronged approach: elicitation sessions with domain experts and a review of relevant literature.

4.1. Modus Operandi

We conducted elicitation sessions with various groups of domain experts, depending on their availabilities and our specific needs. The initial session involved domain experts explicating the specificities and main challenges associated with satellite assembly lines. Next sessions followed another format. At first, based on a selection of previously elicited objectives, model experts presented potential assembly line designs to domain experts. Figure 1 is an example of such a presentation, focusing solely on mandatory goals. Then, model experts proposed potential objectives through drafts of assembly line designs that could be carried out if these goals were to be met. Employing visual representations facilitated domain experts' comprehension of the impact of goals on assembly line designs and enabled them to remain at the appropriate level of description. Both representations served as a springboard for expert discussions. Hypotheses were validated or rejected.

4.2. Model Description

The assembly line performance goal model, depicted in Figure 2, has 34 goals: 8 hard goals and 26 softgoals. In addition, we have specific goals: mandatory goals (in grey on the Figure). These goals could be interconnected through contribution link or correlation link. Contribution links indicate the desired influence of one element on another. Within our model, we utilise two types of contribution: *Help* and *Make*. In addition, we add a correlation link, *Unknown*. As opposed to desired impacts, correlation relationships describe the potential side effects of fulfilling an element [2, 3].

Help relationship is a type of dependency between goals that indicates one goal's achievement or satisfaction contributes positively towards the achievement of another goal. In our model, if the goal *need for new resource is reduced* fulfil, it contributes to the satisfaction of *purchasing new resources is minimised*. It contributes, but it doesn't imply. The *Make* relationship is stronger. Indeed, *Make* relationship between two goals indicates that the successful attainment of one goal is a prerequisite the achievement of the other goal. This relationship implies that the goal being made depends on the accomplishment of the goal that is making it happen. In our model, *only resources already on the assembly are used* obviously implies that *purchasing new resources is minimised*. The *Unknown* correlation indicates the existence of a relationship between goals, but we are unable to determine its nature.

We do not give an exhaustive description of all the goals in our model here. For example, we can focus on goals related to working time (Fig. 3). A *Help* arrow connects the softgoal *workers' shift is minimised* to *workers' working time is optimised*. In this context, a shift refers to time slot assigned to an employee and can vary from 6 to 12 hours. Here, this contribution link signifies that minimising the duration of workers' shifts contributes to optimising their working time. However, achieving the minimum shift duration for all workers does not guarantee the optimisation of

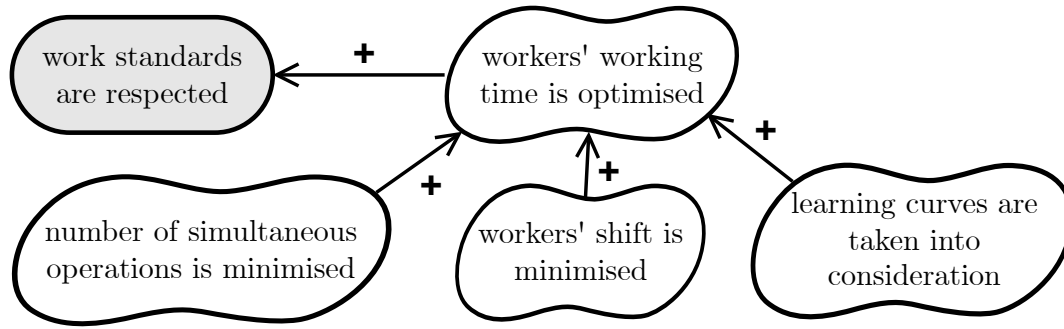


Figure 3: Goals related to work in goal model Figure 2

their working time. Fulfilling other elements, such as minimising the *number of simultaneous operations*, might also be necessary. In our work, we only used *Unknown* correlations, represented by two-way dashed arrows. In Figure 2 there is only one correlation link between *workers' shift is minimised* and *need for new resources is reduced*. This link implies that pursuing the satisfaction of the softgoal *workers' shift is minimised* with the aim of optimising workers' working time might lead to a decrease in the satisfaction of need for *new resources*. Conversely, trying to fulfil the softgoal *need for new resources*, might also decrease the satisfaction of *workers' shift is minimised*. Due to the unquantifiable nature of this impact, we employ the unknown qualifier for the link.

Despite the knowledge of the experts, it became clear that there must be links, which were not elicited. Given the complexity of our system, it seems impossible to find all the relations between objectives using human expertise alone. As we will see in the next session, alternative methods, such as automated calculus, were needed to establish these links.

5. AUTOMATIC CALCULUS TO FIND POSSIBLE INTERACTIONS

The previously presented goal model facilitates the conceptualisation of an *ideal* assembly line design, *i.e.* an assembly line that satisfies all the established goals. However, the realisation of this ideal design might be hindered by the presence of conflicting goals. Achieving complete goal satisfaction within a singular design might algorithmically difficult to obtain or even infeasible due to inherent conflicts or the complexity associated with such designs. Identifying these conflicts offers several advantages : the selection of consistent group of goals we can decline in requirements of the assembly line, the identification of specific goals that need further refinement from domain experts, potentially involving preference ordering and trade-offs and the identification of groups of goals that necessitate the development of specialised tools to facilitate their integration within the assembly line design.

5.1. Approach

We have built an operational research algorithm to generate high-level assembly line designs. This program computes designs that satisfy a selected set of objectives formed by combinations of goals from the model. If the program rapidly computes a design, we infer the absence of conflicts between the given goals. Conversely, if the program requires more than a predetermined time limit to find a solution, or fails to find one altogether, we conclude that the objectives concerned are likely to influence each other. In such cases, a correlation link is added between the involved goals within the model. It is important to acknowledge that the time limit employed for correlation

detection is arbitrarily defined and can be adjusted based on the desired level of accuracy. We consider finding a design within two hours to be *fast* and exceeding one day to be *slow*, but domain experts said that even several days might be considered as *fast* in assembly line design.

5.2. Practical Uses of our Approach

For our application, we have datasets provided by a satellite manufacturer. These datasets encompass the assembly processes for three distinct satellite types (Type 1, Type 2 and Type 3), involving 147 resources. Each satellite requires 40 to 90 individual operations for its construction and there is a 18 month planning horizon. We have developed a constraint programming algorithm. Studied goals within the model is translated into a corresponding constraint within the algorithm. These constraints can be enabled or disabled depending on the specific set of goals we intend to evaluate. The algorithm receives a dataset description as input and outputs a detailed assembly line description. This description includes the operation schedule for each satellite along with the required time and resources. We utilise two constraint satisfaction solvers: CpOptimizer 20.1 [11] and ACE with PyCSP3 [12, 13].

5.2.1 Finding Possible Interactions Between Mandatory Goals

Our first application focused on satisfying the mandatory goals highlighted in grey within Figure 2 and Figure 4. These goals represent essential requirements that must be fulfilled in the final assembly line design. The model comprises four mandatory goals.

- *All satellites are built*, four Type 1, five Type 2, and two Type 3 satellites, must be fully assembled by the end of the 18-month horizon.
- *Satellite construction complies with construction standards*, the construction process of satellites must be compliant to operations needed, their duration and order. The machines and tools needed by the operation have to be available.
- *Work standards are respected*, the process must comply with regulations of the Labour Code, like daily working hours, breaks, holidays, and ergonomic considerations for the workforce.
- *Machines and tools are used according to their specifications*, the utilisation of machines and tools must respect their designated specifications. This includes respecting the number of simultaneous operations allowed on a single machine/tool, as well as machine setup times.

By considering these four mandatory goals, the developed algorithm successfully generated a functional assembly line design in 19.55 seconds. This rapid computation time suggests the absence of negative correlation links between these specific goals.

This initial application is a starting point. We can now add other goals and analysing the resulting assembly line design and computation times.

5.2.2 Finding Possible Interactions Between Delivery Dates and Resources Limitation

A second application aimed to evaluate the impact of introducing two additional goals:

- *Goals delivery dates are met*, the construction of each satellite is achieved before their respective delivery deadlines. Earlier completion is allowed.
- *Only resources already on the assembly line are used* the assembly line design only utilises resources that are already available on it.

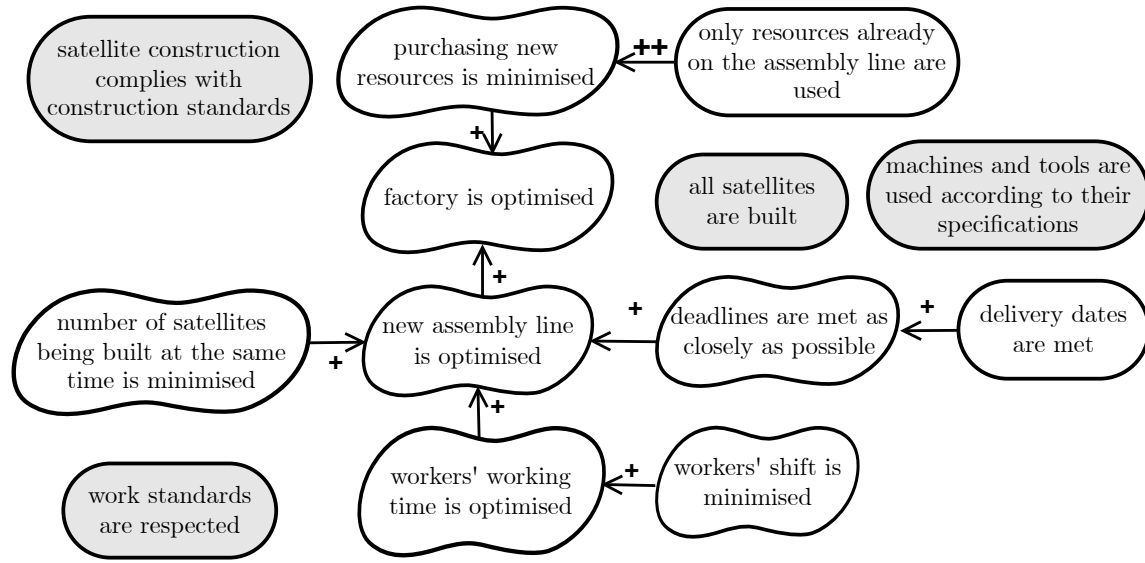


Figure 4: Goals from Fig. 2 evaluated in Section 5

These goals were chosen because they represent the first goals elicited by the domain experts and these constraints are frequently encountered in Operational Research [8]. By including these two additional goals with the four mandatory ones, the algorithm was able to generate an assembly line design in 174 seconds (less than 3 minutes). Figure 1 shows the resulting schedule. From this illustration we can notice that the design ensures that all satellites are constructed with the appropriate number of bars associated at each type. The schedule respects all delivery deadlines. For instance, in the case of Type 1 satellites, the first horizontal yellow bar ends before the first vertical delivery date, which means that the satellite is built before its delivery date, the second bar ends before the second vertical delivery due date, etc. The rapid computation time of 174 seconds suggests that there are likely no significant negative correlation links between these six goals.

5.2.3 Workers' shift

A third application consisted to evaluated mandatory goals with the softgoal *workers' shift is minimised*. To model this softgoal, we fixed a maximum daily working shift of 6 hours for each employee and each day. The algorithm was unable to return a feasible assembly line design. This outcome indicates that our definition of minimal worker shifts might be incompatible with satisfying all mandatory requirements. To express this conflict, we introduced correlation links between the *workers' shift is minimised* softgoal and the four mandatory goals within the model. This means that there is a need for a trade-off between *workers' shift is minimised* and the mandatory goals.

5.2.4 Simultaneous satellite building, resources and worker' shift

The fourth application studied the impact of the three goals, *Goals delivery dates are met*, *number of satellites being built at the same time is minimised* and *need for new resources is reduced*.

- *Number of satellites being built at the same time is minimised*, the number of satellites being built simultaneously on the assembly line has to be reduce.

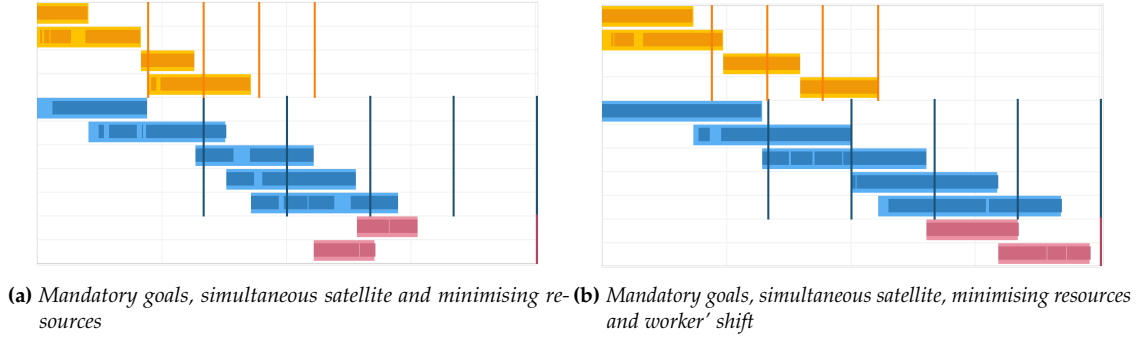


Figure 5: Schedule of assembly line

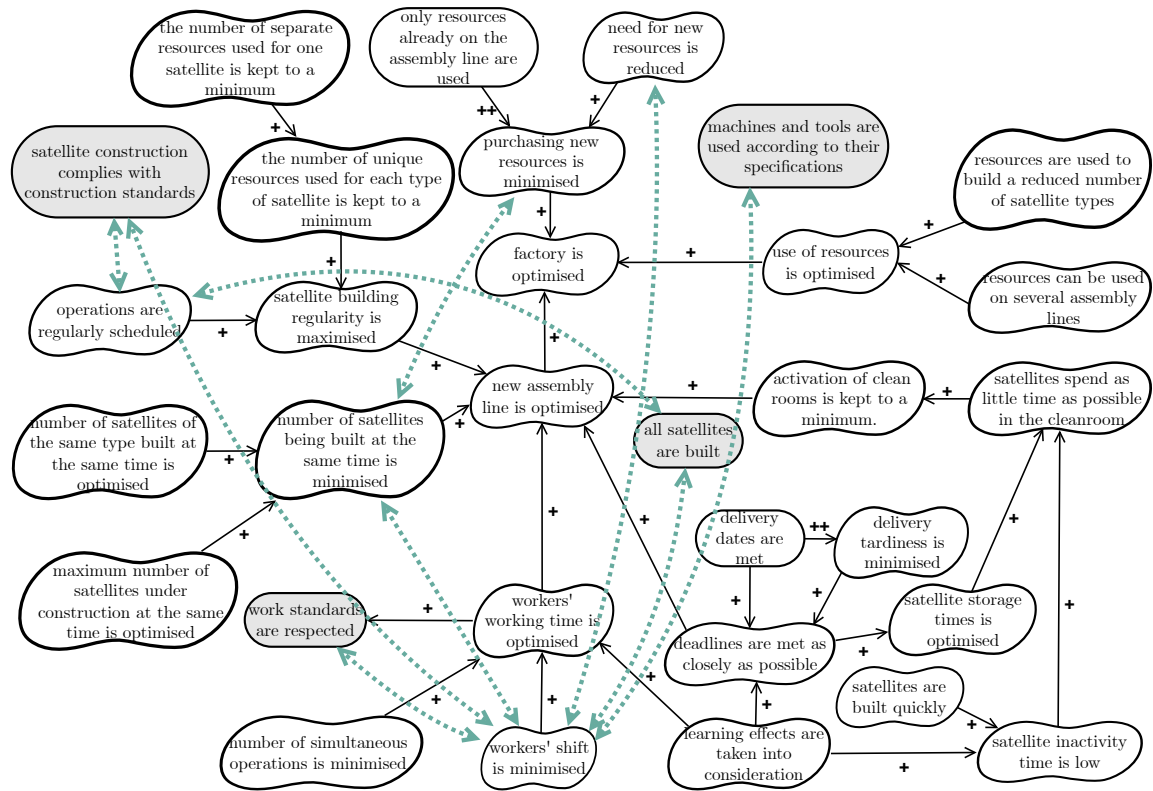


Figure 6: Goal model of factory performance with automatically calculated correlation links

- *Need for new resources is reduced*, the utilisation of existing resources is prioritised and the need to acquire additional resources for the assembly line is minimised.

The algorithm successfully generated an assembly line design within 13.5 seconds when considering these two goals with the four mandatory goals. This rapid calculation implies the absence of negative correlation links between these six goals. The associated schedule is illustrated Figure 5a.

Then we integrated the *workers' shift is minimised* softgoal, the computation time needed around 4 days, 17 hours, and 30 minutes (Figure 5b). The increase in computation time suggests that the satisfaction of the softgoal is likely to have an impact on the achievement of the six other goals. We have therefore added appropriate correlation links to the model to express these conflicts.

We carried out more applications in order to add the correlation links that we see in Figure 6. It should be noted that not all the goals have been evaluated and that our choice of modelling also has an impact on the result obtained. The goal model obtained is therefore just one of the potential goals models we could obtain.

6. RELATED WORK

The field of Goal-Oriented Requirement Engineering (GORE) offers several examples where researchers have combined goal models with computational approaches to achieve optimal solutions. These approaches address complex system design problems by establishing a set of desired goals and using computational methods to identify solutions that best satisfy these goals.

Propagation methods, as described in [10, 1, 7], play a key role in evaluating the impact of achieving specific goals on strategic goals. Each relationship within the goal model is associated with a propagation function, which determines how the satisfaction level of one element influences another. However, potential conflicts can arise during the design process. To address these conflicts, researchers have defined specific rules. For instance, in the context of Non-Functional Requirements (NFRs), [14] proposes a method where satisfaction states are ordered. And, for example, elements receiving conflicting contributions from an and-refinement (all subgoals need to be satisfied) are assigned the lowest satisfaction level.

Solver-based approaches offer another avenue for exploring optimal solutions. In [16], Nguyen combines goal models with Satisfiability Modulo Theories and Optimisation Modulo Theories to identify the maximal set of goals that can be simultaneously satisfied. Similarly, Horkoff and Yu present a method in [9] for defining feasible and optimal solutions based on an iStar goal model and a SAT solver. Baatartogtokh *et al.* in [4] model goal satisfaction as a *Constraint Satisfaction Problem (CSP)*. They then integrate this model with the *Evolving Intension Framework* propagation method to evaluate element state labels based on various objective functions. This approach makes it possible to identify solutions that not only satisfy the greatest number of goals in a specific goal model, but also optimise the set of satisfied goals based on different criteria.

Furthermore, [17] by Sumesh and Krishna proposes a method for achieving an optimal set of goals for the system under consideration. Their definition of optimality not only maximising the number of satisfied goals, but also ensuring the robustness of the final solution. This robustness is evaluated by assigning fuzzy values to both the propagation links and the goal satisfaction levels, and then employing sensitivity analysis techniques. These techniques assess the impact of slight modifications in goal satisfaction on the overall solution, allowing for a more nuanced understanding of the trade-offs involved. However, it is important to note that the information obtained from sensitivity analysis is currently not fed back into the goal model itself.

7. DISCUSSION AND CONCLUSION

In this article, we have explored the conception of a versatile satellite assembly line through the lens of goal-requirement modeling techniques. Our investigation aimed to identify potential conflicts between different goals and assess their impact on the overall efficiency and effectiveness of the assembly line. While the industrial data-based approach has provided valuable insights, it is important to acknowledge its limitations. One drawback is that the scope of conflicts identified may be confined to the specific context of the data, potentially overlooking broader systemic issues or external factors influencing factory performance. However, the industrial data-based approach offers a pragmatic and contextually relevant method for eliciting and understanding real conflicts. Another limitation arises from the fact that the inability to computationally find a model satisfying two conflicting goals does not formally demonstrate the existence of a problem. Although constraint programming aids in identifying conflicts and assessing their feasibility, the absence of a feasible solution does not conclusively prove inherent goal conflicts. Future work should aim to optimise our constraint programming approach to ensure efficient resolution within an acceptable time frame. We could further explore how different elements impact each other by employing more refined computational techniques like sensitive analysis. This analysis could lead to a quantitative or more nuanced qualification of the relationships between these elements.

In conclusion, leveraging computational techniques to analyse goals enables stakeholders and requirements engineers to gain insight into goal relationships, dependencies, and potential conflicts. By identifying inconsistencies and exploring alternative solutions, these techniques facilitate the development of coherent, consistent, and stakeholder-aligned requirements. This holistic approach is essential for optimising the design and operation of versatile satellite assembly lines and ensuring their effectiveness in meeting stakeholder objectives.

REFERENCES

- [1] Daniel Amyot, Sepideh Ghanavati, Jennifer Horkoff, Gunter Mussbacher, Liam Peyton, and Eric Yu. Evaluating goal models within the goal-oriented requirement language. *Int. J. Intell. Syst.*, 25:841–877, 08 2010.
- [2] Daniel Amyot, Jennifer Horkoff, Daniel Gross, and Gunter Mussbacher. A lightweight GRL profile for i* modeling. In Carlos A. Heuser and Günther Pernul, editors, *Advances in Conceptual Modeling - Challenging Perspectives, ER 2009 Workshops CoMoL, ETheCoM, FP-UML, MOST-ONISW, QoIS, RIGiM, SeCoGIS, Proceedings*, volume 5833 of *Lecture Notes in Computer Science*, pages 254–264. Springer, 2009.
- [3] Daniel Amyot and Gunter Mussbacher. User requirements notation: The first ten years, the next ten years (invited paper). *J. Softw.*, 6(5):747–768, 05 2011.
- [4] Yesugen Baatartogtokh, Irene Foster, and Alicia M. Grubb. Visualizations for user-supported state space exploration of goal models. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 281–286, 09 2023.
- [5] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*, volume 5 of *International Series in Software Engineering*. Springer, 2000.
- [6] Fabiano Dalpiaz, Xavier Franch, and Jennifer Horkoff. istar 2.0 language guide. *CoRR*, abs/1605.07767, 2016.

- [7] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005. Agent-oriented Software Development.
- [8] Sönke Hartmann and Dirk Briskorn. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *Eur. J. Oper. Res.*, 297(1):1–14, 2022.
- [9] Jennifer Horkoff and Eric Siu-Kwong Yu. Finding solutions in goal models: An interactive backward reasoning approach. In Jeffrey Parsons, Motoshi Saeki, Peretz Shoval, Carson C. Woo, and Yair Wand, editors, *Conceptual Modeling - ER 2010, 29th International Conference on Conceptual Modeling, Vancouver, BC, Canada, November 1-4, 2010. Proceedings*, volume 6412 of *Lecture Notes in Computer Science*, pages 59–75. Springer, 11 2010.
- [10] Jennifer Horkoff and Eric Siu-Kwong Yu. Comparison and evaluation of goal-oriented satisfaction analysis techniques. *Requir. Eng.*, 18(3):199–222, 09 2013.
- [11] Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Petr Vilím. Ibm ilog cp optimizer for scheduling: 20+ years of scheduling with constraints at ibm/ilog. *Constraints*, 23:210–250, 03 2018.
- [12] Christophe Lecoutre. ACE, PyCSP3-XCSP3, 12 2023.
- [13] Christophe Lecoutre and Nicolas Szczepanski. Pycsp3: Modeling combinatorial constrained problems in python, 2020.
- [14] John Mylopoulos, Lawrence Chung, and Brian A. Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Trans. Software Eng.*, 18(6):483–497, 1992.
- [15] John Mylopoulos, Lawrence Chung, and Eric S. K. Yu. From object-oriented to goal-oriented requirements analysis. *Commun. ACM*, 42(1):31–37, 1999.
- [16] Chi May Nguyen. *Efficient Reasoning with Constrained Goal Models*. PhD thesis, University of Trento, 04 2017.
- [17] Sreenithya Sumesh and Aneesh Krishna. Sensitivity analysis of conflicting goals in the i^* goal model. *Comput. J.*, 65(6):1434–1460, 2022.
- [18] Alex van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *5e IEEE International Symposium on Requirements Engineering*, pages 249–262, 8 2001.
- [19] Eric Siu-Kwong Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering*, pages 226–235. IEEE, 1997.