KEYWORDS collaborative planning and scheduling collaborative project management interventions in large scale facilities

# AN ENHANCED PLANNING and scheduling approach suited to the requirements of COLLABORATIVE PROJECT MANAGEMENT

ABSTRACT

Collaborations imply interdisciplinary work, and require exchanges, communication and compromise. When managing a project, collaboration will lead to complex interactions and feedback between tasks. The planning and scheduling phase of a project already benefits from a large number of tools, mostly based on the Precedence Diagramming Methods (PDM) and its precedence links. This linear vision of how a project shall be planned and scheduled does not fit with the consequences of collaborative work, and unfortunately, no mainstream method for project planning and scheduling does. This work proposes a collaborative planning and scheduling framework based on gathering and handling of temporal constraints through a qualitative temporal algebra, and then on matrix based task-sequence optimization. It provides equal treatment to all constraints, highlights conflicts and propagates the effect of a constraint modification into the existing plan, thus taking coupling, feedback and rework into account.

Mathieu Baudin
European Organization for Nuclear Research (CERN), Geneva, Switzerland, Laboratoire de Conception de Produits et Innovation, ENSAM ParisTech, Paris, France
mathieu.baudin@cern.ch

Pierre Bonnal
European Organization for Nuclear Research (CERN), Geneva, Switzerland
pierre.bonnal@cern.ch

Bertrand Nicquevert
European Organization for Nuclear Research (CERN), Geneva, Switzerland
bertrand.nicquevert@cern.ch

Jean-Michel Ruiz
Laboratoire de Conception de Produits et Innovation, ENSAM ParisTech, Paris, France, École Centrale Marseille, Marseille, France
jean-michel.ruiz@centrale-marseille.fr

## INTRODUCTION

Large scale facilities such as nuclear power plants, chemical plants, particle accelerator facilities such as the ones present at CERN require the work of many different specialists in many different scientific fields, from the technicians to the engineers and sometimes researchers. In addition, personnel members at CERN come from more than forty different countries. Consequently, projects run for these large scale facilities will require the participation of all these different professionals to be successful.

All these large scale facilities are highly collaborative environments. Project management in this specific context shall be more complex than in smaller businesses with defined fields of action, even though a project will always imply some level of collaboration. To our knowledge, project management lacks tools and methods suitable for such situations, especially when it comes to planning and scheduling.

In fact, methods used today still rely on methods which can date as far back as the 1960s, like the Precedence Diagramming Method, which have already proved to be more than useful. But it can be shown that such methods, typically displaying projects as linear endeavors, are not entirely suited to the complex interactions *(e.g., coupling, feedback, loops, etc.)* that can appear between tasks in highly collaborative contexts.

Luckily, methods enabling to take such characteristics into account exist, in project planing and scheduling as well as in process modeling. It is in particular the case of the Design Structure Matrix, as matrix-based framework specializing in handling dependencies, loops and coupling. But temporal information about such dependencies must also be gathered as close as possible to the needs of the collaborators before being fed to planning and scheduling tools, and suitable frameworks also lack in project management, the mostly used system being the four dependencies used in the Precedence Diagramming Method.

It can be shown that project management could benefit from frameworks used so far in planning and scheduling for artificial intelligence, and more specifically from a framework introduced in the 1980s by James Allen: Allen's interval algebra, a qualitative temporal algebra using an interval-based representation of time.

This article is divided in five sections. A more formal definition of collaboration and of the collaborative planning and scheduling context considered is given in section one, and the classic methods used for planning and scheduling in project management are described in section two, where their limitations in collaborative contexts are also analyzed. Section three focuses on Allen's interval algebra and DSM, before proposing the collaborative DSM, a collaborative planning and scheduling framework combining them. Section four describes two practical cases for illustration purposes, and section five provides the conclusions of this work, as well as future research works.

## 1. What is Collaborative Planning and Scheduling?

### 1.1 Defining the Collaboration

In this paper, planning and scheduling methods suited to collaborative project management will be considered. Consequently, it is important to formally define our understanding of a collaboration, and of a collaborative planning and scheduling environment: a collaboration is the association of several entities, performing different roles in order to work together to reach one or several goals.

The word "entities", used to refer to the different collaborators, has been chosen carefully: a collaboration can be as simple as that of a rider and his horse, and as complex as those existing at CERN with several hundreds of individuals performing very diverse and distinct activities in several dozens of fields. It can also involve machines, and one can think for instance about multi-agent systems or about human-computer collaboration in artificial intelligence. This paper will mainly focus on human-human collaborations, and is based on case studies of CERN and its collaborative engineering projects. Regarding the notion of a common goal, we do not consider it to be the main motivation for collaborators to join the collaboration, but rather the motivation behind the existence of the collaboration itself. In addition, any collaborator may have their own personal goals to reach, as long as these do not conflict with the aim which is common to all.

When mentioning the roles played by the collaborators, the emphasis is placed on the fact that these roles are distinct one from another. At CERN, in order to run, maintain and upgrade the particle accelerators, more than forty different disciplines are involved: among many others, cryogenics engineers, magnets specialists, beam engineers or radiation protection experts, not to mention particle physicists. As such, a collaboration stems from a need of different types of input to reach the aforementioned common goal, and from the fact that one collaborator cannot answer to this need alone. Or, in other words, we understand the collaboration to be necessarily interdisciplinary, and not the association of several identical entities, thus producing "inter-laced knowledge" *(Boisot, Nordberg, Yami, & Nicquevert, 2011).*

Collaboration is more than just "adding people up". Collaborating is not about juxtaposing different actors and managing their independence. All these different actors have their own practices, and in a way their own languages, even if they all happen to speak English together every day. Working in a collaboration implies trying to understand the people one is working with: what added value their work gives to the collaboration, and what their possible limitations are. As a result, managing a collaboration is about coordinating all those different actors while leaving them enough autonomy in their own jobs, by defining and managing "coordinated zones of autonomy" through "exchange spaces" *(Nicquevert, 2013; Nicquevert, Boujut, et al., 2013).* These spaces, much like trading zones *(Galison, 1997),* stand at the interface defined between different fields. They aim to establish common references and tools, while leaving everyone enough freedom to do their own job in their zone of autonomy. In this perspective, the framework of communities of practices can also be introduced *(Wenger, 1998) (see* **Figure 1**), where the project manager is the "broker" who will help practitioners to create links between different disciplines and handle conflicts when necessary. Other notions, such as the "boundary object" *(Star, 2010),* serve as references or common definitions for collaborators at an interface.

### 1.2 Collaborative Planning and Scheduling

Every project goes through a planning and scheduling phase, to estimate the duration of activities, set deadlines for milestones and deliverables and estimate its overall duration. Collaborators must get a precise idea of temporal links that exist between the activities in which they participate and the rest of the project. They must be able to communicate their constraints, and understand those of others. The planning and scheduling part of a collaborative project is thus a temporal interface in a collaboration, where all collaborators exchange the temporal information between the activities of the project. The tools and methods used in this interface are typically planning and scheduling systems aiming at producing schedules, for instance, under the form of Gantt charts. However, in our collaborative context, the whole process of elaborating such schedules should fulfill precise collaboration requirements.

The context of planning and scheduling at CERN for the interventions in particle accelerator facilities *(Baudin, Bonnal, & De Jonghe, 2012)* displays most of the issues we want to address in collaborative planning and scheduling: collaborations which plan and schedule several thousands of interventions every year, with several hundreds of collaborators involved. These collaborators range from the technicians who perform the work, or the group-leaders who need the intervention, to intervention planning coordinators in charge of the overall planning process. All these collaborators have constraints of their own and need to specify them: when working with others, precedence links appear between the tasks *(e.g., a transportation task to a given location in a tunnel might be necessary before installing a piece of equipment, and a radiation survey will be required before sending personnel into a radioactive area, etc.)* and resource constraints also exist. Human resources are limited, as well as space in underground facilities and equipment which become unavailable during an intervention. It becomes important to enable the collaborators to express these constraints in a clear and understandable way for everyone, and even to highlight possible conflicts before deploying coordination efforts.

Indeed, coordination is needed, otherwise a fully collaborative planning system of such a size, with total equality between the requests, will quickly become chaotic. On the other hand, collaborators need this equal treatment for their requests to some extent, otherwise the planning system is no longer collaborative: if only a few individuals collect all the temporal constraints and establish the schedule without interaction with any of the other participants, the collaborative elaboration of the schedule is by-passed and there is no exchange at the temporal interface. As a result, it becomes necessary to strike a balance between those two extreme situations.

## 2. Limitations of Usual Planning and Scheduling Methods in a Collaborative Environment

The following section reviews classic planning and scheduling methods *(the Critical Path Method (CPM)* and the Precedence Diagramming Method *(PDM))* and describes their limitations when facing a collaborative context, before proposing alternative ones based on artificial intelligence techniques to handle temporal constraints in such situations. Hints on how to detect and work on conflicts between temporal constraints, even before a coordination phase, shall be provided, and a framework able to handle coupling and feedback of tasks when automatic resolution of such conflicts is not possible, shall be proposed.

### 2.1 Usual planning and Scheduling Methods: CPM and PDM

The CPM and the PDM were developed during the 1950s and 1960s *(Kelley & Walker, 1959) (Fondahl, 1962) (Crandall, 1973)*. They were among the first analytic methods of project planning and scheduling to be developed and are still used as a base for many project scheduling software tools. Both of those methods use the Graph Theory to analyze the activity network linked to a project.

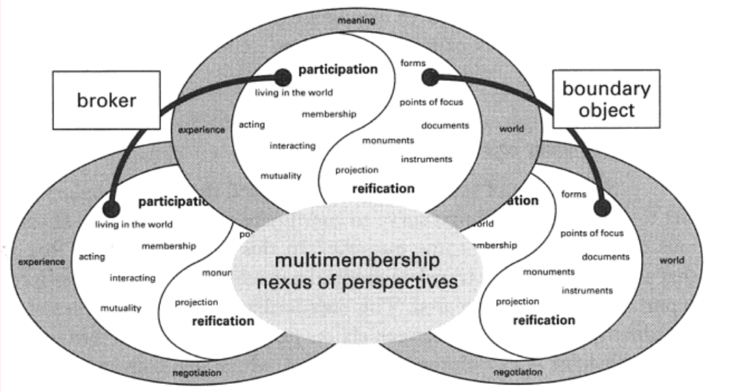


**FIGURE 1.** *Representation of an interface illustrating the notion of broker and boundary object. Figure cited from (Wenger, 1998).*

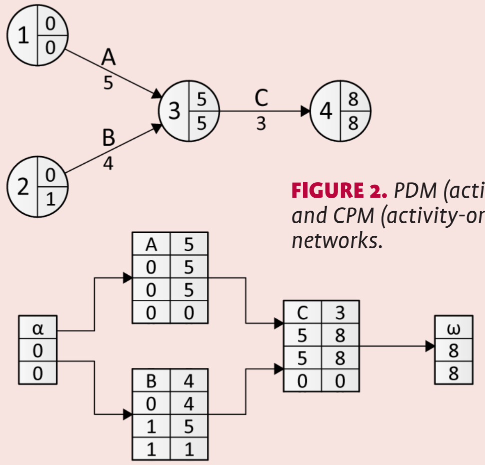


**FIGURE 2.** *PDM (activity-on-arc) and CPM (activity-on-node) networks.*

Their graphical formalism is very simple. On the one hand, the CPM graph is usually found under an activity-on-arrow diagram. On the other hand, the PDM method is mostly represented using the activity-on node formalism, using the arrows as constraints *(see* **Figure 2***)*. The main difference between the two models lies in how the precedence constraints are handled. While the CPM only handles finish-to-start *(FS)* constraints between activities, the PDM deals with start-to-start *(SS)*, start-to-finish *(SF)* and finish-to-finish *(FF)* constraints as well. In addition, PDM also allows the implementation of delays between activities. To simplify the description and not be subject to the difference between the two formalisms, we will now refer to each time point corresponding to the start or the end of any activity as an event.

Each activity of the graph is associated to one value: its duration. The user then needs to calculate four additional values: its earliest and latest start, and its earliest and latest finish. The calculation is the same for both CPM and PDM. It consists of a three step method based on propagation, using the duration of each activity. First, the earliest date of each event is calculated in the forward way, from the initial node to the terminal one. The latest date is then calculated the same way by starting from the terminal node and going backward to the initial one. The activities for which the earliest and latest dates coincide are part of the "Critical Path" *(and named "critical activities")*, which means that the sum of their duration *(plus the duration of delays for PDM)* defines the duration of the whole project. For the activities which are off the critical path, a third steps enables the user to calculate their "float", indicating the delays they can be subject to without affecting the global duration of the project.

CPM and PDM have been popular since their development. Many other planning and scheduling methods use them as a base, like PERT *(Malcolm, Rosenbloom, Clark, & Fazar, 1959)*, which follows the same forward-pass and backward-pass calculation procedure, but adds a stochastic dimension to the calculations through the assignment of a random distribution of duration to the activities. Likewise, many Resource Constrained Planning and Scheduling Problems *(RC-PSP)* are solved using a first step of PDM, and then apply resource leveling algorithms *(see for instance (Pinedo, 2008)* or *(Brucker, Drexl, Möhring, Neumann, & Pesch, 1999))*. PDM is also still used nowadays in most of the commercial planning and scheduling tools: even if the calculations differ a little from the standard algorithm described previously, the set of constraints used by planning and scheduling professionals is still exactly the same one used in PDM *(as a reminder: SF, FS, SS and FF)*.

Unfortunately, we will see that there are two major disadvantages in using PDM-based methods when planning and scheduling projects in a collaboration. The first lies in the need for clarity in the representation of a workflow generated by a collaboration, with all the couplings and interactions that can appear between tasks in such a complex planning environment. The second, and most important in our eyes, is a lack of expressiveness of the set of constraints. We will show that collaborative submissions of temporal constraints can lead to situations that cannot be handled using a PDM scheme, with requests that are not possible to translate in PDM and conflicts that cannot be represented and solved.

### 2.2 Process Modeling vs. Planning and Scheduling

As pointed out in the definition of collaborative planning and scheduling context, collaborators are working in an interface, which holds communication and exchange as a fundamental basis. As a consequence, it is difficult to think that the output of the collaboration will be deterministic and linear as it appears in CPM and PDM. A discussion implies possible conflicts, decisions to make and sometimes a rework or modification of a task after its original definition. Collaborative planning and scheduling, as other collaborative endeavors *(design processes for instance)* is complex, and a collaborative plan or schedule shall be better modeled dynamically as explained by B. Nicquevert *(Nicquevert, 2013, chap. 9)*, or by Repenning and Sterman *(Repenning & Sterman, 2001)*.

This is why notations such as loops, but also conjunctions or disjunctions, in addition to the sequences which are the only available links in CPM and PDM, would be very useful to gain in expressiveness and point out the dynamic links between the different steps in the plan. For instance, the Decision Box planning method *(Eisner, 1962)* uses conjunction and disjunction to model scientific research projects, acknowledging the fact that some paths might be preferred to others, which should be ignored as a result. As for the loops, the collaborative planning and scheduling can benefit from process modeling methods based on workflow, such as Business Process Modeling and Notation *(BPMN) (Object Management Group (OMG)*, 2011) which include rework, loops or multiple instances of tasks.

As we will later show, the matrix-based method of Design Structure Matrix *(DSM)*, initially used for system analysis, can be applied to project management, and addresses the notions of coupling and feedback particularly well.

### 2.3 Taking Collaborative Constraints into Account

Finally, we want to point out one more limitation that appears in frameworks based on CPM and PDM in collaborative planning and scheduling environments. We observed in the field that, when trying to gather requests from collaborators concerning temporal constraints between the tasks they were in charge of and other tasks of a project, we could find ourselves in situations which were not easy or even possible to translate into the set of constraints available in PDM.

Let us consider two activities A and B. The four temporal constraints handled in PDM have the following meanings:

- Finish-to-start: *A FS B* means "B cannot start before A finishes";
- Start-to-finish: *A SF B* means "B cannot finish before A starts";
- Start-to-start: *A SS B* means "B cannot start before A starts";
- Finish-to-finish: *A FF B* means "B cannot finish before A finishes".

If we define the start and finish dates for activity A as $A_s$ and $A_f$ respectively, and do the same for B, each of those four constraints shall give multiple possible planning solutions, the smallest number being two possibilities resulting from FS with $A_f = Bs$ or $A_f < B_s$ *(and obviously $A_s < Af$ and $B_s < B_f$ to have activities with a positive duration)*. Now considering **Figure 3**, one can see that eight possible configurations can result from each of them, depending on the relative positions of the constrained points *($B_s$ and $A_s$ for SS and $B_f$ and $A_f$ for FF)*. These possibilities are represented by the grey bars at the bottom of the figures. The symbols displayed next to them belong to Allen's Interval Algebra, a temporal representation that shall be defined later in the article and will be used in the framework we propose. The curious reader can find their meaning in **Table 1**. Similar calculations give a total of twelve possibilities for the SF constraint.

It is thus understandable that this restricted set of constraints is actually hiding an important complexity, and it is in the end relatively difficult to translate precise requests that one might encounter in a collaborative constraint-gathering phase. For instance at CERN, the most common request made by collaborators is that two activities should not take place together at any point of their execution[1]. This apparently simple request actually has no trivial translation in PDM constraints. And at first sight, it is also the case for other precise requests aiming at:

- having two tasks to start or end at the same time ($A_s = B_s$ or $A_f = B_f$);
- having two "equal" tasks, starting and ending together ($A_s = B_s$ and $A_f = B_f$);
- having a task contained in (or containing) one other ($A_s < B_s$ and $B_f < A_f$).

We shall see later that this is entirely not true, as only the initial set of four constraints do not allow for these cases. However, we shall later show that summing or intersecting the set of planning and scheduling solutions of several precedence links leads to satisfactory solutions, though less precise and clear to the user than what they originally requested. And we will see that clearer sets of constraints exist, that can easily overcome such problems.

So as we can see, the set of PDM constraints covers a lot of situations, but not in the clearest way possible. Indeed, one precedence link corresponds to multiple possible outcomes. And reciprocally, when collaborators request planning and scheduling constraints, they come with concrete requests: the tasks should be equal, or start together, or be distinct in the plan. This implies concrete relations between time points similar to the ones we used previously. But conversions and combinations are needed to adapt to PDM-compliant constraints. If we then go back to the definition of collaborative planning and scheduling as the temporal interface within a collaboration, we need to place the emphasis on exchange, communication and understanding between different collaborators. This lack of clarity *(and sometimes of feasibility)* concerning how the temporal constraints are handled may well lead to problems of understanding and frustration from the users who will need to interact in real time with others to establish satisfactory plans and schedules. In fact, this process implies discussions and compromises, so that modifications of some constraints are likely to occur. When such simple concepts as "activities are distinct", or "A occurs during B" imply combinations of several notations *(that all users may not be familiar with)*, this collaborative process will at best be slow and painful, and in worse cases be faulty.

# 3. Proposing a Collaboration Compliant Planning and Scheduling Framework

The basis of the framework we will now develop has already been successively laid out in *(Baudin et al., 2012)* and *(Baudin, Bonnal, & Ruiz, 2013)*. To address the limitations encountered by classic planning and scheduling systems in collaborative contexts, we worked on two different axis concerning the temporal constraints on the one hand, and the feedback and loops between tasks generated by collaborative processes on the other. We are able to propose a framework based on two methods which solve the issues exposed previously, though we are fully aware that there may be many different methodologies. After a quick description of the two methods it combines, which are Allen's interval algebra and the DSM, and explanations on the reasons of their selection, we will describe the Collaborative DSM framework combining the two.

## 3.1 Allen's Interval Algebra

### *3.1.1 Definitions and Basic Algorithms*

To fit with the described requirements, one can find a framework which possesses a larger set of constraints than PDM and detects conflicting constraints by looking into the artificial intelligence planning field. More specifically, those two characteristics led us to consider Allen's interval algebra, a calculus introduced in the early 1980s by James Allen *(Allen, 1983)*. It suits collaborative planning and scheduling in several ways: first, like CPM and PDM, it gathers the precedence links between the tasks, but in a simpler way, using lower-level constraints. It can then be used to order the foreseen tasks following those constraints. And finally, it can take into account resource or logistics constraints on top of the sequential *(also called potential)* constraints mentioned previously. By doing so, it can even be considered as a technique that bridges the gap between planning and scheduling *(Smith, Frank, & Jonsson, 2000)*.

Allen's interval algebra proposes an interval based representation of time, as opposed to a point based *(typically with dates)* representation. The tasks or events are then described by the interval of time during which they occur. And since intervals have a duration, it is still possible to switch back to point-based representation using the starting and ending points of time intervals. However, it is better suited than the latter to describe situations in which relative knowledge of time and events are involved. For instance, if we take the morning schedule of a person, it will be easier to say that they had breakfast, then went out to take the bus, opened a newspaper while waiting, and finished reading while commuting, than trying to describe all these events, some of which overlap each other using precise time points. Finally, the time interval representation makes it easier for a project manager to deal with the lack of precision that often characterizes engineering projects: engineers and technicians, when confronted with the question of duration will hardly answer with a precise duration, and even less with start and finish dates; they will more likely use fuzzy time descriptors *("about a week", a "few hours", etc.)* that will be easier to handle using a time-interval-based notation and a propagation of precedence constraints, than precise starting and ending dates.

Now that we have described the time interval formalism, let us proceed to the temporal constraints between tasks. These temporal constraints define an ordering of the time intervals, taking into account potential and logistics constraints. In Allen's interval algebra, there are seven temporal constraints *(before, meets, during, overlaps, starts, finishes and equals)* to which are added their inverses, which gives a total of thirteen relations, equals being its own inverse *(see **Table 1**)*. Given a network of tasks represented by time intervals, it is then possible to assign one or a vector of several of these thirteen relations to each pair of intervals to describe the constraints between the tasks they represent[2]. This set of time intervals and temporal constraints is referred to as a temporal constraint network *(TCN)*. TCNs have been widely studied in artificial intelligence in the 1980s and 1990s, mostly from a computation-theoretic point of view *(see for instance (Allen, 1983), (Vilain, Kautz, & van Beek, 1989), (van Beek & Cohen, 1989)*, and *(van Beek, 1990))*. More recently, a few publications have, like us, applied those constraint networks to processes, to model business processes *(Lu, Sadiq, Padmanabhan, & Governatori, 2006)* or workflows *(Dufour-Lussier, Le Ber, & Lieber, 2012)*.

There are several types of problems that can be solved starting from an initial TCN, and the most crucial one is to compute the transitive closure of a given set of constraints, when a new interval constraint is appended in the network. It aims at determining, when it exists, the minimal set of constraints between all pairs of intervals in the network, discarding the values which are not part of a consistent scenario. This problem can be referred to as the minimal labeling problem. And unfortunately, it is a NP-Hard problem, which implies that exact solutions can only be computed by algorithms that behave exponentially. This is a major interest for studies in the computation theoretic field, and many optimized exponential algorithms *(van Beek, 1990)(Valdes-Perez, 1987)*, special cases, sub-algebras *(Krokhin, Jeavons, & Jonsson, 2003)* and approximation algorithms *(van Beek & Cohen, 1989)* have been studied in this context.

The algorithm used by Allen to compute the transitive closure is one of these approximation algorithms. It is a propagation algorithm *(see **Figure 4**)*, which, as its name indicates, propagates the effects of a modification of one or several constraints. It maintains a queue of pairs <i, j> of intervals for which the constraints have been modified, and for each pair it determines whether the new defined relation for <i, j> constrains the relation between any other pair <i, k> or <k, j>, where k is any interval of the network distinct from i and j. It should be noted that in the previously cited references, the absence of knowledge of the relations between two intervals will be indicated by assigning the pair a vector containing all the thirteen relations, thus indicating that any of them could be true. Conversely, an empty set of constraints assigned to a pair of intervals during propagation will show that the network contains conflicting constraints.

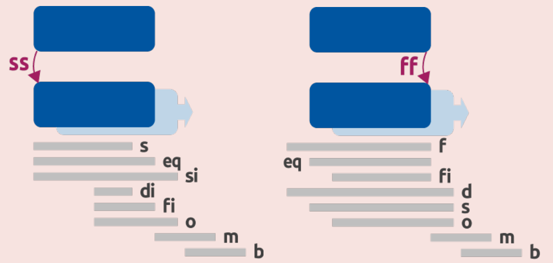


**FIGURE 3.** *The different planning configurations for two activities constrained with start-to-start and finish-to-finish relations.*

As we previously indicated, we are solving this NP-hard problem with an approximation algorithm. As shown by Allen *(Allen, 1983)*, it will never infer a wrong constraint, but may fail to notice conflicts, as only three-node-sub-networks are tested *(going to larger sub-networks actually induces the exponential behavior)*. However, as indicated in *(Vilain et al., 1989)*, the propagation algorithm we described remains interesting to use in databases where the constraints are mostly used to record the temporal relations between events, but not so much to compute new ones from the initial set. And this is exactly where we stand, as we consider an environment of interacting users who want to submit tasks into a collaborating system and be noticed if their scheduling constraints happen to be in conflict. They only have little interest in inferring additional relations. We will now show how they could benefit from the use of Allen's interval algebra, and more specifically from the set of constraints it provides in terms of expressive power,

[1] Our thanks to Katy Foraz (CERN, EN-MEF-OSS) of the Intervention Management Working Group for this private communication.

[2] All the relations of the algebra are mutually exclusive, so a vector of temporal relations corresponds to the disjunction of all possible relations for the considered pair of time intervals.

and in terms of equal treatment of the constraints.

### 3.1.2 Strengths in Collaborative Environments

Let us now show some of the strengths Allen's interval algebra displays in collaborative environment. The first is, in our opinion, its enhanced set of constraints and the very practical way of handling them using the propagation algorithm. In fact, we mentioned while covering CPM and PDM that all of the four precedence links commonly used in classic planning and scheduling methods *(FS, SF, SS and FF)* did not answer to precise requests, but rather led to several planning and scheduling options. And in fact, if we take the definitions of those four precedence relations, we can translate them into Allen's interval algebra notation and get the four following equations:

$$A\ FS\ B = A < b,m > B \tag{1}$$

$$ASS\ B = A < b,m,o,di,s,si,fi,e > B \tag{2}$$

$$A\ FF\ B = A < b,m, o,d, s, f, fi, e > B \tag{3}$$

$$A\ SF\ B = A < b,m, mi, o,oi, d, di,s,si, f, fi, e > B \tag{4}$$

Consequently, it will be very straightforward to specify precise temporal constraints, especially those requesting equality *(meets, starts, finishes and equals)* using Allen's interval algebra while combinations shall be necessary in the classic PDM scheme. For instance, the combination of two PDM constraints are needed to express that two activities have to be consecutive with no delay *(relation "meets" in Allen's interval algebra)* as indicated in equation 5.

$$A < m > B = A\ FS\ B \cap B\ SF\ A \tag{5}$$

To reach this result, one has to consider that if a relation is true in Allen's interval algebra, its reciprocal relation holds as well. For instance:

$B < m > A \Leftrightarrow A < mi > B$

Then, an intersection using the sets given in equations 1 to 4 is performed. Similarly, one can find an equivalent for the Allen's interval algebra "starts" and "finishes" by intersecting two PDM relations as shown in equations 6 and 7. However, we can see that we do not find precisely A<s>B or A<f>B, but a set of solutions where either $A_s = B_s$ or $A_f = B_f$, associated to other possible solutions including equality. The same can be shown for the "during" relation *(see equation 8).* To be thorough, we should finally mention the cases of equality and separation we noted back in section 3.3. One can actually achieve A<e>B using PDM precedence links, but by combining four of them: it can be done by intersecting the results of equations 6 and 7, or the two results in equations 8.

$$A\ SS\ B \cup B\ SS\ A = A < s, si, e > B \tag{6}$$

$$A\ FF\ B \cup B\ FF\ A = A < f, fi, e > B \tag{7}$$

$$\begin{aligned} A\ FF\ B \cup B\ SS\ A &= A < d, f, e > B \\ A\ SS\ B \cup B\ FF\ A &= A < di, fi, e > B \end{aligned} \tag{8}$$

Finally, the case of activity separation *(A distinct from B)* is equivalent in both representations: it has no specific notation in Allen's interval algebra and is the disjunction of the four relations <b, bi, m, mi> while it is obtained in PDM precedence links using $A\ FS\ B \cup B\ FS\ A$. However, we have seen that the intersection of PDM constraints fails to reproduce individually every relation present in Allen's interval algebra, which can be used to translate very straightforward request in a simple and powerful representation.

In addition to what we consider to be a step forward in terms of expressiveness and clarity, we should also highlight the fact that in Allen's interval algebra, constraints are handled equally thanks to the propagation algorithm. Every time a new constraint is added into the TCN, it is checked against all previously entered constraints and every potential conflict can be detected, regardless of which constraint was first entered into the network, and which collaborator submitted it. This should be regarded as major interest for a collaborative environment, considering that coordination or moderation shall be performed afterward. Every user can submit constraints in a clear and expressive way, and then observe potential conflicts caused by their request prior to any decision making.

Finally, it was previously mentioned that the interval based representation of time can be used to deal with sequential and logistics constraints. This is traditionally not the case with CPM and PDM, which usually require further steps of resource leveling after planning. We shall later see that collaborative intervention planning and scheduling can need to represent transport and travel tasks as much as the main skeleton of activities to perform, especially in large-scale facilities.

Along those strengths in collaborative environments, a problem remains, which is to handle and solve the conflicts. In fact, when the propagation algorithm returns a failure in a computer science problem, the set of constraints is deemed inconsistent. And in a computer science problem, "no solution" is a solution. This is however a luxury the project manager cannot afford. When a project is launched and its activities identified, they need to be scheduled, regardless of the consistency of the collaborators' requests. Compromises are needed, and the schedule needs to keep track of all the complex interactions and possible conflicts generated by the collaborative planning and scheduling processes. Allen's algebra does not provide such possibilities, and it is for such purposes that we could benefit from process modeling capabilities as mentioned in 3.2, and in particular of looping and coupling.

Some authors have already studied ways to draw Allen's interval algebra and process modeling closer, by defining intermediate formalisms enhancing inter-interoperability between the two *(Dufour-Lussier et al., 2012).*

On our side, we do not need disjunction or conjunction as much as coupling, since our interest lies more in the representation of conflicts, which tend to generate feedback and loops between tasks which will be scheduled. This is why we studied the possibility of coupling Allen's interval algebra with a framework that specializes in dependency, looping and feedback modeling: the DSM.

### 3.2 DSM

DSM stands for Design Structure Matrix, or Dependency Structure Matrix. It is a matrix-based framework introduced in the early 1980s *(Steward, 1981)* by Donald Steward which enables representation in a compact manner and the optimization of sequences of tasks in processes or structures of elements in decompositions of systems. We will only focus on sequences of tasks in this paper, and thus only on the former case, i.e., the activity DSM.

Its main strength lies in the representation it provides for dependencies between tasks, and more specifically its ability to detect loops and feedback in a process. The DSM is in fact a squared matrix, with each row *(or column)* representing an activity in the sequence. Its content can then be viewed as binary: if the DSM element $\alpha_{ij}$ is a mark *(typically an "X")*, or is set to one, it indicates a dependency between activity i and activity j, while a blank or a zero shows that no link exists between them.

The precedence link is read in accordance with the chosen convention. In this paper, we will use the input-on-rows, output-on-columns convention, meaning that a mark on one task's row represents the input it receives from other tasks, and the marks placed on its column represent its output to the other tasks. In this convention, if one reads the matrix as a sequence of activities *(from top to bottom and from left to right)*, the marks appearing under the diagonal represent information transferred to tasks performed later in the sequence, while the ones appearing above the diagonal indicate a feedback to previous tasks, or even a loop in case of inter-dependency.

Once a DSM is built, the aim is then to optimize its sequence of activity to avoid all feedback if possible, or minimize it if necessary. The algorithm used to proceed is called partitioning. In our convention, a successfully partitioned DSM will be lower triangular, or with one occasional mark just above the diagonal in case of a loop, as shown in the example in **Figure 6**. If loops exist, a tearing algorithm can be used to determine the optimal order of the activities in each loop. Going into the details of these already well described algorithms is not in the scope of this paper, and we provide more information on partitioning in *(Baudin et al., 2013)*, and a larger list of methods is available in *(Gebala & Eppinger, 1991).*

It should be noted that other versions of DSM exist, such as the Numerical DSM which numerically quantifies the strength of the dependencies or many different types of parametrized DSM, which use the non-zero elements to describe the dependencies more in depth than the basic binary DSM. The framework we propose later in this paper is in fact a new parametrized DSM based on information gathered from users using Allen's interval algebra for the dependencies and estimated duration of activities. However, whatever the type of DSM used, the optimization algorithms will still process the DSM as a binary matrix.

### 3.3 The Collaborative DSM: Methods and Algorithms

The framework we propose is based on a new parametrized DSM containing Allen's interval algebra information instead of marks or ones in the nonzero elements. The initial DSM can contain multiple temporal relations per non-zero matrix element. However, in opposition to traditional TCN solving *(Lu et al., 2006)*, we leave the matrix element empty instead of assigning all the thirteen relations to pairs of activities on which we have no information regarding temporal constraints, thus leaving blank matrix elements, and enabling the performance of DSM partitioning in the same way it is done for binary DSMs.

The methodology follows six steps:

- step 1: submit new constraints;
- step 2: propagate the effects of the new constraints (Allen's propagation algorithm, Figure 4);
- step 3 (optional): look for compromises if conflicts are signaled;

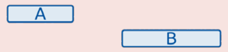
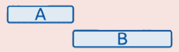
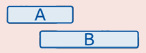
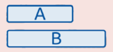
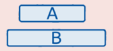
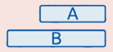
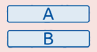
| Relation | Interpretation | Gantt chart-like illustration |
|---|---|---|
| A b B<br>B bi A | A takes place before B<br>B takes place after A |  |
| A m B<br>B mi A | A meets B<br>B is met by A |  |
| A o B<br>B oi A | A overlaps B<br>B is overlaped by A |  |
| A s B<br>B si A | A starts with B<br>B is started with A |  |
| A d B<br>B di A | A is during B<br>B contains A |  |
| A f B<br>B fi A | A finishes with B<br>B is finished by A |  |
| A = B | A equals B |  |

**TABLE 1.** *Table of the thirteen relations of Allen's Interval Algebra.*

- step 4: solve the TCN;
- step 5: generate as many DSMs as existing solutions;
- step 6: partition DSMs and compare results.

Step 1: one or several new constraints can be added. It is also possible to reduce the number of possible relations in an already existing vector of temporal relations for any already constrained pair of intervals.

Step 2: for this step, Allen's propagation algorithm is used in order to look for possible effects of the newly entered constraints, usually a reduction of connected constraint-vectors. It is not necessary to infer all minimal relations between every pair of intervals in the network, since the DSM only focuses on direct precedence links between tasks. Thus, the use of the approximation algorithm is well justified, since most of the work performed by the algorithm consists of a verification of the transitivity rules and is a task of polynomial complexity.

Step 3: this step is only performed if one or several conflicts is diagnosed in the previous one. So far, more work remains to be done and this may be the subject of further publications. Currently, the compromises are obtained by relaxing temporal constraints when possible. For instance, if a relation A<m>B is causing a conflict, the equality $A_f = B_s$ is relaxed into $A_f < B_s$ or $A_f > B_s$. If one or several of the relations obtained through relaxation *(in this case A<b>B and A<o>B)* do not belong to the original set of relations, and at the same time satisfy the transitivity rules of the

```
While (Queue not empty):
    For each pair $< i,j >$ in Queue:
        For each $k \neq j$ Comparable to $i$:
            $Temp = < i,k > \cap Transitivity(< i,j >, < j,k >)$
            if $Temp = \emptyset$
                then signal contradiction
            else if $Temp \neq < i,k >$
                then $< i,k > = Temp$
                Append $< i,k >$ to Queue

        For each $k \neq i$ Comparable to $j$:
            $Temp = < k,j > \cap Transitivity(< k,i >, < i,j >)$
            if $Temp = \emptyset$
                Then signal contradiction
            else if $Temp \neq < k,j >$
                then $< k,j > = Temp$
                Append $< k,j >$ to Queue
```

**FIGURE 4.** *James Allen's constraint propagation algorithm. The function Transitivity(<i, j>, <j, k>) returns the set $r_3$ of all possible relations between i and k knowing the relations $r_1$ between i and j and $r_2$ between j and k. A summary table of these rules is given by J. Allen in (Allen, 1983).*

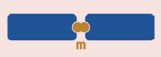
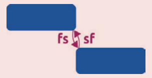
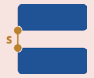
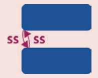
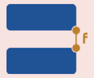
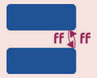
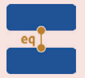
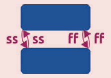
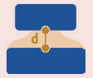
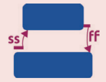
| Allen's interval algebra relations | Equivalent PDM Combinations |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**TABLE 2.** *Illustration of equations 5 to 8, representing the needed combinations of PDM precedence links to reproduce single Allen's interval algebra constraints. The case of the "equals" relation has also been added.*

conflicting triplet of activities, it is possible to proceed to step 4.

Step 4: aims at determining all possible scenarios based on the results of previous steps, by finding all instances consisting only of singleton relations between pairs. This step boils down to solving a Constraint Satisfaction Problem (CSP), restricted to the pairs of intervals for which temporal information is available. This is performed by trying the different singulet combinations and propagating the effects using the same algorithm as in step 2. Scenarios with only singulet labeling for all non-zero DSM elements are deemed "consistent", and represent potential instances of projects.

Step 5: now that collaborative constraint gathering is performed, the collaborative DSM *(actually one per consistent scenario)* can be built. This step is quite complex, since most situations described in Allen's interval algebra involve coupling and impose the adjacency of the involved interval in the process, as shown in **Figure 7**. In addition, potential failure of steps 3 and 4 have not been mentioned yet and need to be taken into account when building collaborative DSMs. If a conflict arises and no compromise is found, the tasks involved in the conflict should be considered together in the schedule *(thus coupled or in a loop in the DSM)*, and a coordination effort deployed afterward to solve the issues for each case. Like the compromising algorithm, this step of the framework is still under development and shall be further described in later works. However, as shown in **Figure 9**, one possible solution is to use numerical values standing for delays between connected tasks and to keep "X" marks to indicate conflicts. This has the advantage of preparing duration calculations, but the inconvenience of asking the collaborators for precise qualitative information they do not necessarily have, as we insisted on the fact that Allen's interval algebra aims at describing relative knowledge on events and time. Another possibility would be to partition the collaborative DSM as it is *(and thus to skip this step 5)*. In this case, the reciprocal relations should be indicated to signal the couplings: for instance, if $DSM_j = \mu$, then $DSM_j = \mu\iota$.

Step 6 finalizes the procedure with the partitioning of all the collaborative DSMs and additional calculations. At this step, the total duration of all the possible sequences previously determined can be calculated, and the choice is left to the project manager as to which solution to implement if a choice persists. Other criteria can play a role in this decision making step. Taking the example of intervention planning and scheduling at CERN, longer interventions associated with lower radiation doses received by the personnel can be preferred to shorter solutions leading to higher doses.

# 4. Aplication to a Few Use-Cases

This collaborative DSM framework was initially developed to enhance collaborative intervention planning and scheduling in the facilities at CERN. These are large scientific facilities dedicated to running and colliding particle beams in particle accelerators *(the most famous being the Large Hadron Col-lider (LHC))* which are located in underground tunnels. Since the operation of particle accelerators produces ionizing radiations, everything is done to keep human interventions underground to a minimum. However, it is not avoidable, and maintenance operations necessitate the exposure of personnel. This is why enhanced interventions are designed to keep radiation exposures as low as reasonably achievable. The examples that are developed in this section are based on this context.

## 4.1 What Is an Intervention?

All human interventions in beam facilities are considered. There are five main types to distinguish:

- preventive maintenance: replacement of a piece of equipment before failure;
- corrective maintenance: replacement of a piece of equipment after failure;
- inspections: typically radiological surveys or quality controls;
- equipment or facility consolidation;
- equipment or facility upgrade.

The three former categories are more operations oriented, and are performed on repetitive occasions, during short technical stops or short planned shutdown of the facilities. The latter two are one of a kind interventions aimed at improving the performances of the facilities as well as their reliability, availability, maintainability and safety *(RAMS)* for further operations. They can necessitate longer shutdowns such as the one currently occurring at CERN until the end of 2014.

Because of their unavoidable collaborative nature and the variety of tasks and constraints they require, which is largely due to the complexity of the environment they are performed in, we consider interventions in CERN's facilities to be small projects, and an interesting illustration for our framework. The symbols displayed in **Table 3** are the ones which shall be encountered in the following examples.

## 5.2 A Simple Illustration

We will start with a very simple process, described in **Figure 10**. One human resource named 'Gaston performs one preventive maintenance task A on a piece of equipment M10, in a location Z6. To do so, the logistics tasks have to be included: traveling to Z6 from departure point D1 through locations Z3, Z4 and Z5 before performing A, and from Z6 to D1 after A is completed. In **Figure 10**, the collaborative DSM is displayed on the left. As for CPM and PDM, a dummy task 00 is created to represent the starting point of the intervention. The constraint <m, b> is set, since Gaston can leave for Z6 right away or have some delay. However, the constraint <m> is requested between the travel and A, and then between A and the return trip to minimize the time spent in the facilities, especially if Gaston is exposed to radiations.

For comparison, the matrix on the right displays the information traditionally available with a CPM-PDM scheme: the tasks linked to logistics are not taken into account in these models, and only the dummy start and the maintenance task A appear with a finish-to-start constraint.

So far, no collaborative work has been performed to plan and schedule this example. Only one human resource is involved, the sequence of tasks is straightforward *(partitioning is not even needed)*. The only collaborative interaction we could foresee here would be to have the Radiation Protection Expert *(RPE)* check the radiation conditions in all different locations involved prior to the intervention. The process diagram would be the one displayed in **Figure 11**. The resulting DSM would be almost the same, with an additional line and column inserted before the travel task, and both *"before"* and *"meets"* relations acceptable between the survey and the travel. Here again, no partitioning is needed and the sequence of activities is straightforward.

More complexity can be achieved just for this simple intervention. In fact, the preventive maintenance task could necessitate heavy new equipment, and require transport. A second intervention team to handle cabling is usually required, and radiation safety is not the only safety check performed on site. Fire, electrical or cryogenic hazards can necessitate pre-intervention inspections and marking. The simple linear intervention described in 10 will now look like 12 and includes no less than ten tasks, at least five different human resources *(counting the safety inspections)* and can mobilize up to five different locations in the facilities since the transport may be large and slow.

Such a large process diagram could benefit from a more compact representation. And when planning the intervention, it is likely that all these involved collaborators may request incompatible planning and scheduling situations. Handling such cases is the main objective of the collaborative DSM.

## 4.3 Replacing a Beam Instrument

This second example is more complex and will serve to illustrate a conflict in terms of submissions. The aim of the modeled intervention is to change a faulty beam instrument. This intervention can be performed in seven steps:

- 1a: Disconnect the instrumentation signal cables;
- 1b: Disconnect the associated controls signal cables;
- 2: Replace the beam instrument;
- 3: Replace the associated controls;
- 4a: Reconnect instrumentation signal cables;
- 4b: Reconnect the controls signal cables;
- 5: Test the new system.

Three collaborators are involved. The first is in charge of the instrumentation, the second in charge of the control blocks and the third is responsible for the electrical connections. They shall be designated respectively by the letters I, C and E. Collaborators I and C are both required to work on task 5, 'test the system. The intervention is planned using the collaborative DSM framework,

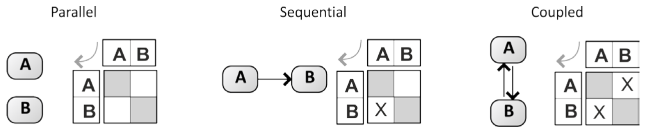


**FIGURE 5.** *The dependencies in the binary DSM.*

omitting for simplification purposes the logistics that should later be included as indicated in the first example. This results in three submission, one from each participant, as shown in **Figures 12** to **14**.

The first submission by collaborator I is straightforward, since the schedule is created with two tasks: the dummy start and task 2, the replacement of the instrument. The second submission, by collaborator C appends two tasks to the plan: tasks 3 and 5. The controls could be replaced before the instrument, but C requests *2<b>3*, to be able to perform the tests right away *(3<m>5)* and minimize the time in the facilities.

The third submission is made by collaborator E (**Figure 14**). It adds cable related activities to the schedule: disconnecting the signal cables both for the beam instrument and its associated control blocks before the replacement tasks *(tasks 1a and 1b)*, and reconnecting them after the replacements and before the tests take place. The added constraints are *1a<b, m>2, 1b<b, m>3, 2<b, m>4a, 3<b, m>4b, 4a<b, m>5, and 4b<b, m>5*. Consequently, a conflict will be found by the propagation algorithm between the constraints *3<m>5, 3<b, m>4b and 4b<b, m>5*, as the tests cannot be performed right after the replacement of the control blocks *(as requested previously by C)* if the cables have not been reconnected. The collaborators need to rework on this plan, and relaxation of constraint *3<m>5* into *3<b>5* is enough to solve the conflict, since it allows for the insertion of task 4b between tasks 3 and 5.

This example is theoretical, and some tasks still need to be added to reach a realistic description. In fact, two travel tasks per collaborator should appear, to reach the replacement location and to travel back. As in the previous example, a transportation task for the new equipment could be needed, and several safety checking steps for radiation protection and other features are likely to be required.

Also, this example was designed to illustrate the concept of conflict and compromise. But in practice, conflicts will most likely not appear during the planning and scheduling of the main skeleton of tasks, but rather when assigning resources. In fact, human resources can be subject to several assignments, in other interventions. And at CERN, the limited available space underground can be considered as a resource *(the LHC tunnel has a width of about 8.0 m, and much space is occupied by the particle accelerator and various pieces of equipment)*. Consequently, it may not be possible to transport the new piece of equipment through a location already occupied for other purposes, hence a spatial resource conflict. This explains why the most common request for temporal constraints at CERN is "A distinct from B".

| ↩ | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | ▨ | | | X | X | | |
| B | | ▨ | X | | | X | X |
| C | | | ▨ | X | | | X |
| D | X | | | ▨ | | | |
| E | | | | | ▨ | | |
| F | | | X | | | ▨ | |
| G | | | | X | | X | ▨ |

**INITIAL DSM**

| ↩ | E | A | D | C | F | G | B |
|---|---|---|---|---|---|---|---|
| E | ▨ | | | | | | |
| A | X | ▨ | X | | | | |
| D | | X | ▨ | | | | |
| C | | | X | ▨ | | X | |
| F | | | | X | ▨ | | |
| G | | | X | | X | ▨ | |
| B | | | | X | X | X | ▨ |

**PARTITIONED DSM**

**FIGURE 6.** *An initial DSM and the matrix obtained after partitioning.*

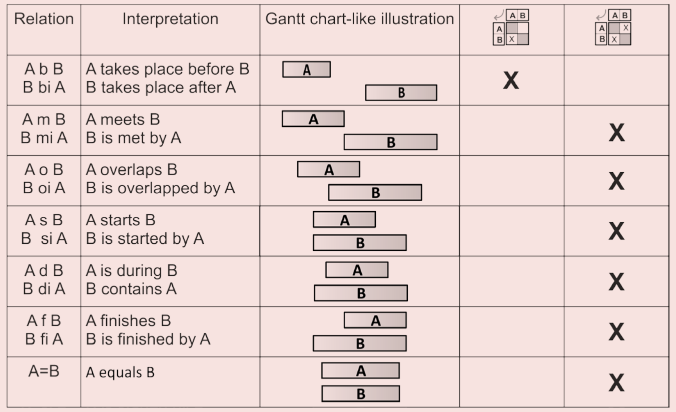
## 5. Conclusions

A definition of a collaboration, and then of a collaborative planning and scheduling environment has been given. It is a context involving several different collaborators, specializing in many different disciplines, to work towards a common goal. In terms of project management, the goal is to successfully realize a project. These collaborative structures generate complex interactions, which makes exchanges and communication crucial. Consequently, it is necessary to define "interfaces", or "exchange zones" where the collaborators can communicate using the same terms and understand each other, even if they have very different backgrounds. When it comes to planning and scheduling the activities in a project, we are in what can be called the "temporal interface" of a collaboration. All collaborators need to exchange temporal information linking their activities in the project, to plan and schedule it in a way that fulfills all constraints, or a least a maximum.

To do so, a collaborative planning and scheduling framework has been described: the collaborative DSM. Coupling Allen's interval algebra and DSM, it enables different collaborators to gather temporal constraints using an expressive set of constraints, to check their compatibility, to work on compromises in case of conflict and to optimize the sequence of activities. It handles both potential and logistics constraints, and treats them equally in the propagation algorithm, which is an asset for collaborative work: all submissions are treated the same way regardless of the order in which they are submitted, and none is given priority over another. Its compact DSM-based representation enables to represent large sequences of tasks, including transportation and logistics. These latter are traditionally not included in classic planning and scheduling methods, but are of major importance in large scale facilities such as CERN and its underground installations subject to ionizing radiations, but also large industrial facilities *(e.g., nuclear power stations, chemical plants, etc.)* where risk and safety are at the core of every project or intervention.

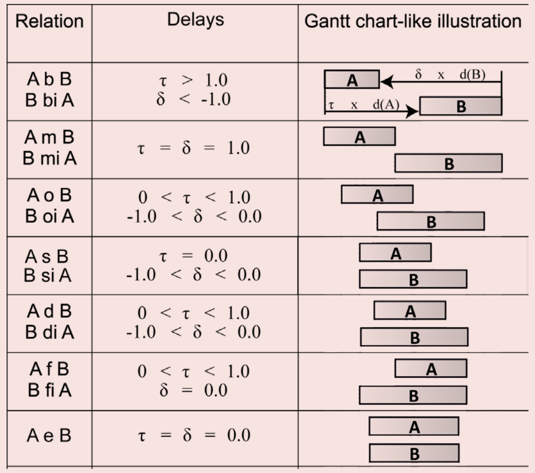
| | |
|---|---|
| | Corrective maintenance |
| | Preventive maintenance |
| | Movement, travel |
| | Transportation task (loaded)<br>Transportation travel (empty) |
| | Inspection task |

**TABLE 3.** *The process symbols used in this paper to model interventions.*



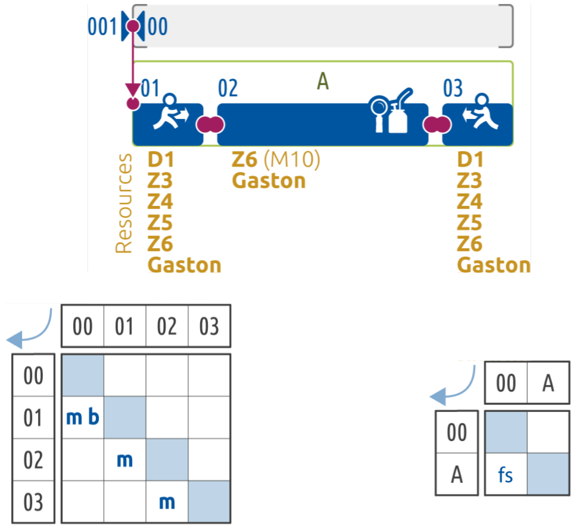
| Relation | Interpretation | Gantt chart-like illustration | | |
|---|---|---|---|---|
| A b B<br>B bi A | A takes place before B<br>B takes place after A | A<br>B | X | |
| A m B<br>B mi A | A meets B<br>B is met by A | A<br>B | | X |
| A o B<br>B oi A | A overlaps B<br>B is overlapped by A | A<br>B | | X |
| A s B<br>B si A | A starts B<br>B is started by A | A<br>B | | X |
| A d B<br>B di A | A is during B<br>B contains A | A<br>B | | X |
| A f B<br>B fi A | A finishes B<br>B is finished by A | A<br>B | | X |
| A=B | A equals B | A<br>B | | X |

**FIGURE 7.** *Conversion from Allen's interval algebra to binary DSM dependencies.*



| Relation | Delays | Gantt chart-like illustration |
|---|---|---|
| A b B<br>B bi A | $\tau > 1.0$<br>$\delta < -1.0$ | A ← δ x d(B)<br>τ x d(A) → B |
| A m B<br>B mi A | $\tau = \delta = 1.0$ | A<br>B |
| A o B<br>B oi A | $0 < \tau < 1.0$<br>$-1.0 < \delta < 0.0$ | A<br>B |
| A s B<br>B si A | $\tau = 0.0$<br>$-1.0 < \delta < 0.0$ | A<br>B |
| A d B<br>B di A | $0 < \tau < 1.0$<br>$-1.0 < \delta < 0.0$ | A<br>B |
| A f B<br>B fi A | $0 < \tau < 1.0$<br>$\delta = 0.0$ | A<br>B |
| A e B | $\tau = \delta = 0.0$ | A<br>B |

**FIGURE 8.** *Use of delays (noted r and δ) in the DSM instead of Allen's symbols, as defined in the right cell of the first line.*



| ↩ | 00 | 01 | 02 | 03 |
|---|---|---|---|---|
| 00 | ▨ | | | |
| 01 | m b | ▨ | | |
| 02 | | m | ▨ | |
| 03 | | | m | ▨ |

| ↩ | 00 | A |
|---|---|---|
| 00 | ▨ | |
| A | fs | ▨ |

**FIGURE 9.** *A simple intervention with one preventive maintenance task to perform, and two logistics tasks (travel) needed to reach the intervention location.*

The choice of the two methods of interest might seem arbitrary, since many other methods exist in project management and operations science, in order to model processes, handle loops and plan and schedule activities within a project. Future work soon to be published shall focus on the review of many methods of interest and bring out arguments as for the selections of DSM and Allen's interval algebra. The framework proposed so far is still under development, and further research will also focus on several aspects that have been highlighted throughout this paper, such as the improvement of the search for compromises when a conflict is detected, and the search for more efficient ways *(for instance, involving estimated duration of activities)* to express Allen's interval algebra notations into a DSM that will be more straightforward to convert in a Gantt chart-like schedule.

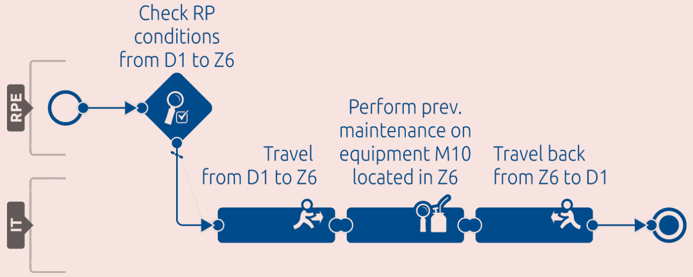

Check RP conditions from D1 to Z6
RPE
IT
Travel from D1 to Z6
Perform prev. maintenance on equipment M10 located in Z6
Travel back from Z6 to D1

**FIGURE 10.** *Same intervention as in Figure 10, with a radiation protection inspection inserted before. IT stands for Intervention Team.*

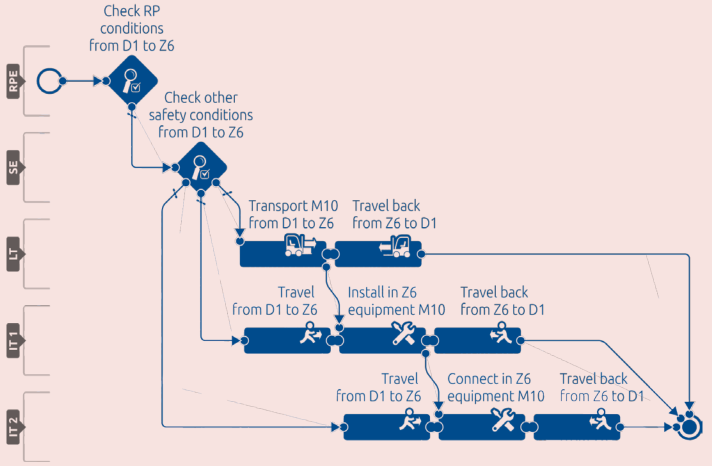
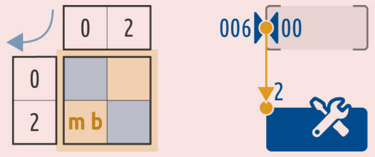
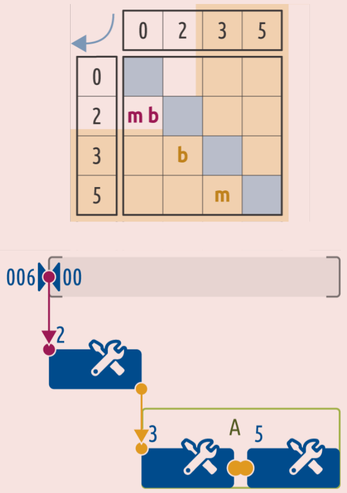

Check RP conditions from D1 to Z6
RPE
Check other safety conditions from D1 to Z6
SE
Transport M10 from D1 to Z6
Travel back from Z6 to D1
LT
Travel from D1 to Z6
Install in Z6 equipment M10
Travel back from Z6 to D1
IT 1
Travel from D1 to Z6
Connect in Z6 equipment M10
Travel back from Z6 to D1
IT 2

**FIGURE 11.** *Same intervention as in Figure 10, including safety inspections, transport of equipment and a second intervention team to perform the cabling.*



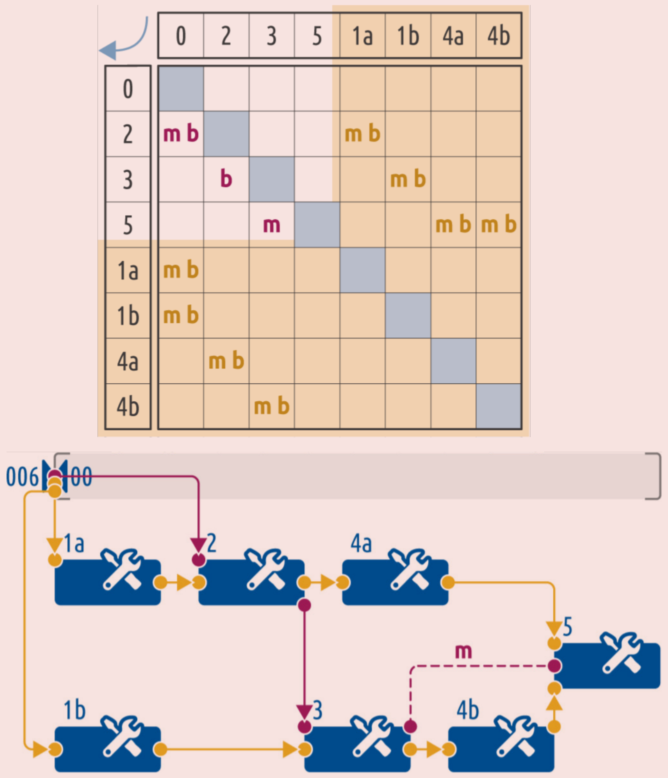
| | 0 | 2 |
|---|---|---|
| 0 | | |
| 2 | m b | |

**FIGURE 12.** *First planning step of beam instrument replacement: the collaborator I in charge of the instrument plans his replacement task (2).*



| | 0 | 2 | 3 | 5 |
|---|---|---|---|---|
| 0 | | | | |
| 2 | m b | | | |
| 3 | | b | | |
| 5 | | | m | |

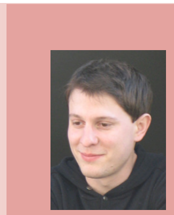
**FIGURE 13.** *Second planning step of beam instrument replacement: the collaborator C in charge of controls plans after task 2 his replacement task (3), and also the test phase (5), requesting immediate succession between the two.*



| | 0 | 2 | 3 | 5 | 1a | 1b | 4a | 4b |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | |
| 2 | m b | | | | m b | | | |
| 3 | | b | | | | m b | | |
| 5 | | | m | | | | m b | m b |
| 1a | m b | | | | | | | |
| 1b | m b | | | | | | | |
| 4a | | m b | | | | | | |
| 4b | | | m b | | | | | |

**FIGURE 14.** *Third planning step of beam instrument replacement: the collaborator E in charge of cabling adds the disconnection (1a and 1b) and reconnection tasks (4a and 4b).*

## authors



**Mathieu Baudin** I was born in 1988, and graduated in 2011 from the French Engineering school Centrale Marseille. This degree is equivalent to a masters degree in engineering. In 2011 as well, I graduated from the University Paul Cézanne, Aix-Marseille III in a masters degree in Theoretical Chemistry, and Analytical Chemistry for which I performed a six month master thesis at the University of Nottingham in translational medical imaging, with Prof. Thomas Meersmann.

I am now performing my PhD at CERN, Geneva, Switzerland, in industrial engineering. My grad school is at Ecole Nationale Supérieure d'Arts et Métiers ParisTech (ENSAM ParisTech, Paris, France). I am working on intervention models suited to collaborative intervention planning and scheduling, and their optimization based on radiation protection data.

**Bertrand Nicquevert** European Organization for Nuclear Research (CERN), Geneva, Switzerland.

**Jean-Michel Ruiz** Laboratoire de Conception de Produits et Innovation, ENSAM ParisTech, Paris, France, École Centrale Marseille, Marseille, France.

**Pierre Bonnal** Dr. Pierre Bonnal holds engineering degrees in mechanical engineering from the l'École nationale d'ingénieurs de Tarbes, France, and in industrial engineering from the École centrale Marseille, France, a M.Sc. in project management from the University of Quebec at Trois-Rivières in Canada and a Ph.D. in industrial management from the Institut national polytechnique de Toulouse, France.Pierre claims a solid industrial background including work with Airbus Industrie (F); Fatigue Technology Inc. (USA); Doris Engineering (F), Areva-SGN (F) as well as more than sixteen years with CERN (CH) mainly involved with the LHC project management. In 2008, he benefited of a two-year sabbatical to join the University of Applied Sciences of Western Switzerland-Geneva as a professor and dean of the Department of Business Administration. As from September 2010, he is back to CERN, involved in project management support to the many studies and projects on-going at CERN. Pierre is also the CERN's scientist-in-charge of a European funded initiative that aims at developing enhanced means (processes, hardware systems and software) for preventing human intervention for increased safety in infrastructures emitting ionizing radiation. Pierre is a former chairperson of the IPMA affiliated Swiss Project Management Association of Western Switzerland and a usual senior lecturer in project management and R&D management with the business schools of the Universities of Geneva and Lausanne in Switzerland.

## BIBLIOGRAPHIC references

**Allen, J. F.** *(1983).* Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26, 832-843.

**Baudin, M., Bonnal, P., & De Jonghe, J.** *(2012).* A DSM Based"2.0" System for Human Intervention Planning and Scheduling in Facilities Emitting Ionizing Radiations. Proceedings of the 14th International Dependency and Structure Modelling Conference, Kyoto, Japan.

**Baudin, M., Bonnal, P., & Ruiz, J.-M.** *(2013).* Combining Activity DSM with Temporal Logic for Collaborative Planning and Scheduling. International Journal of Advanced Robotic Systems, Special Issue: Telerobotics and Systems Engineering for Scientific Facilities.

**Boisot, M., Nordberg, M., Yami, S., & Nicquevert, B.** *(Eds.). (2011).* Collisions and Collaboration: The Organization of Learning in the ATLAS Experiment at the LHC. Oxford University Press.

**Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E.** *(1999).* Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research, 112, 341.

**Crandall, K. C.** *(1973).* Project planning with precedence lead/lag factors. Project Management Quarterly, 6.

**Dufour-Lussier, V., Le Ber, F., & Lieber, J.** *(2012).* Extension du formalisme des flux opérationnels par une algèbre temporelle. Sixièmes Journées de l'Intelligence Artificielle Fondamentale *(JIAF)*, 133-142. Retrieved from http://arxiv.org/abs/1209.5664

**Eisner, H.** *(1962).* A generalized network approach to the planning and scheduling of a research project. Operations Research, 10, 115-125.

**Fondahl, J. W.** *(1962).* A non computer approach to the critical path method. Tech. Report no. 9. Dept of Civil Engineering, Stanford U., Stanford, CA..

**Galison, P.** *(1997).* Image and logic: A material culture of microphysics. University of Chicago Press.

**Gebala, D. A., & Eppinger, S. D.** *(1991).* Methods for analyzing design procedures. Proceedings of 3rd International ASME Conference on Design Theory and Methodology, Miami, Florida.

**Kelley, J. E. J., & Walker, M. R.** *(1959).* Critical-Path Planning and Scheduling. 1959 Proceedings of the Eastern Joint Computer Conference, 160-173.

**Krokhin, A., Jeavons, P., & Jonsson, P.** *(2003).* Reasoning about temporal relations: The tractable subalgebras of allen's interval algebra. Journal of the ACM, 50, 591-640.

**Lu, R., Sadiq, S., Padmanabhan, V., & Governa-tori, G.** *(2006).* Using a Temporal Constraint Network for Businne Process Execution. Seventeenth Australian Database Conference, Hobart, Autralia.

**Malcolm, D. G., Rosenbloom, J. H., Clark, C. E., & Fazar, W.** *(1959).* Application of a technique for R&D program evaluation *(PERT)*. Operations Research, 7, 646-669.

**Nicquevert, B.** *(2013).* Manager l'interface. approche par la complexité du processus collaboratif de conception, d'intégration et de réalisation : modèle transactionnel de l'acteur d'interface et dynamique des espaces d'échanges. Unpublished doctoral dissertation, Grenoble U. *(presented 23 Nov 2012)*

**Nicquevert, B., Boujut, J.-F., et al.** *(2013).* Modeling engineering interfaces in collaborative activities: a transactional model. In Proceeding of the 19th internationale conference on engineering design.

**Object Management Group** *(OMG). (2011).* Business process model and notation *(BPMN)*, version 2.0. Retrieved from http://www.omg .org/spec/BPMN/2.0/

**Pinedo, M. L.** *(2008).* Scheduling: Theory, algorithms, and systems, third edition. Springer.

**Repenning, N. P., & Sterman, J. D.** *(2001).* Nobody ever gets credit for fixing problems that never happened. California management review, 43*(4)*, 6488.

**Smith, D. E., Frank, J., & Jonsson, A. K.** *(2000).* Bridging the gap between planning and scheduling. Knowledge Engineering Review, 15.

**Star, S. L.** *(2010).* This is Not a Boundary Object: Reflections on the Origin of a Concept. Science, Technology & Human Values, 35*(5)*, 601617.

**Steward, D. V.** *(1981).* The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE transactions on Engineering Management, 28.

**Valdes-Perez, R. E.** *(1987).* The Satisfiability of Temporal Constraint Networks. In Aaai *(pp. 256260)*.

**Van Beek, P.** *(1990).* Exact and Approximate Reasoning about Qualitative Temporal Relations. Unpublished doctoral dissertation, Department of Computer Science, University of Waterloo, Ontario, Canada.

**Van Beek, P., & Cohen, R.** *(1989).* Approximation Algorithms for Temporal Reasoning. Department of Computer Science, University of Waterloo, Ontario, Canada.

**Vilain, M., Kautz, H., & van Beek, P.** *(1989).* Constraint Propagation Algorithms for Temporal Reasoning: a Revised Report. Readings in qualitative reasoning about physical systems, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc, 373-381.

**Wenger, E.** *(1998).* Communities of practice: Learning, meaning, and identity. Cambridge University Press.