Timing Nonfunctional Requirements

Ivan J. Jureta and Stéphane Faulkner

PReCISE, University of Namur, Belgium
ivan.jureta@fundp.ac.be; stephane.faulkner@fundp.ac.be

Abstract. Analysis of temporal properties of nonfunctional – i.e., quality – requirements (NFRs) has not received significant attention. In response, this paper introduces basic concepts and techniques needed for the specification and analysis of time properties of NFRs.

1 Introduction

Topic. Software can be characterized by *what* services it provides to stakeholders, and *how well* it provides these services, i.e., dependably, rapidly, and efficiently, among other. The former are usually called *functional* and the latter *nonfunctional* (i.e., quality) requirements [2]. Research and industry seem to agree that several complementary approaches spanning various steps of the software development process and software evaluation are needed to ensure a degree of quality as close as feasible to the ideal levels desired by the stakeholders [3]. Assistance can be provided to the software engineer in the form of dedicated concepts and techniques for the specification and analysis of software nonfunctional requirements. Currently, such help relies to a considerable extent on the theoretical foundations laid out in the established Nonfunctional Requirements (NFR) framework [2] which combines the concept of nonfunctional requirement for representation, with contribution relationships for reasoning about NFRs.

Problem and Contributions. One perspective on setting quality levels and assuring that they are met, which has remained outside the scope of the NFR framework is the specification and analysis of how levels of satisfaction of NFRs vary over time. The framework does not cover the definition and analysis of properties of time intervals over which a particular sastisfaction level is expected of NFRs – instead, the implicit assumption seems to be that every nonfunctional requirement is to be fixed and enforced during the entire duration of software operation or the development and evaluation project. Change of environment conditions, stakeholders' requirements, and conflicts between NFRs make this assumption unrealistic: there certainly are intervals in system operation, during which some NFR is to be favored over other, only to reverse the preferences at some other times. Specifying expected levels of satisfaction *over time* for NFRs helps in detecting interdependencies between NFR satisfaction levels and indicates the need for discovering conditions for the anticipated variation of satisfaction levels. This paper suggests concepts and techniques for introducing the information about time in the specification and analysis of NFRs, that is, to enable temporal properties of NFRs to be rendered explicit and studied.

Running example. To illustrate the suggested concepts and techniques, initial requirements expressed for Air-Traffic Management (ATM) information systems were analyzed. Between 1994 and 1998, the European Organization for the Safety of Air Navigation (Eurocontrol) organized a number of consultation workshops for the aviation community in the aim of documenting these stakeholders' expectations regarding the ATM information systems strategy for the period after the year 2000. Information in workshop deliverables [1] contains actual initial considerations expressed by the various stakeholders, including a large number of quality-related concerns. In the present paper, fragments from the case study are used to exemplify concepts and techniques introduced in the remainder.

2 Timeline, Condition and Policy Concepts

Introducing time in the analysis of nonfunctional concerns changes how the software engineer proceeds to the selection of alternative combinations of satisfaction levels, which in essence amounts to the selection of a particular set of desired software properties among alternatives. Without considering the variation of NFR satisfaction over time, the software engineer selects among static representations of software alternatives. However, when time is accounted for, the software engineer can compare the alternative sets of behaviors of the software in relation to NFRs over time. Static representations of alternatives such as, e.g., NFR goal contribution graphs ([2] and related) cannot provide such comparisons.

Overall, adding time information to an NFR proceeds as follows: A *timeline*, composed of *time points* organized into *intervals* is defined for each NFR; *conditions* for the variation of desired NFR satisfaction levels are given along with *policies* (see below) to apply for ensuring the desired quality leves be met; a condition starts an interval, and the interval is set to last until the change of conditions (timing out can also be a condition, so that specific durations can be accounted for) – the satisfaction level of the NFR is thus assumed stable accross each interval. Timelines of different NFRs are then combined until the entire NFR specification is accounted for in the *general NFR timeline*. As iterations in specification due to incremental acquisition of NFR-related knowledge both by the stakeholders and the software engineer make distributed and computer-assisted NFR specification more likely than a sequential process executed exclusively by the software engineer, it is not appropriate to work on the NFR book timeline without having elaborated individual NFR timelines. Combining timelines is partly automated through the timeline synchronization technique (see, Technique 3) algorithm that generates alternative paths based on trade-offs between satisfaction levels of distinct NFRs. From there on, selecting among alternative timelines is a matter of analysis of the general NFR timeline using techniques described below (§3).

The presence of labels make the timeline a particularly useful means for relating within a single representation the conditions, policies, stakeholders, along with time information, and metric values. This integration, along with timeline merging, and creation and comparison of alternatives are discussed later on

(§3), after the present discussion defines basic timeline conceptualization and the subsequent one introduces the condition and policy concepts.

Definition 1. *A **timeline** $\mathcal{T}_q \stackrel{def}{=} (G_q, F_q)$ for the NFR $q \in \mathcal{Q}$ consists of:*

1. *A graph $G_q = (N_q, E_q, source_q, target_q)$ defined over a set of nodes N_q and multiset (i.e., bag) of edges, with $source_q : E_q \longrightarrow N_q$ and $target_q : E_q \longrightarrow N_q$, which map, resp., an edge to its source node and an edge to its target node.*
2. *A set of functions F_q used to label the graph G_q, as follows:*
   (a) *$timing_q : N_q \longmapsto \mathbb{T}$ is a partial function mapping edges to time points, whenever times can be specified, i.e., are known by the stakeholders or the engineer. Intervals can be derived for successor edges, and interval duration by substituting the left from right endpoint, provided the two are known. The time domain is introduced in Def.2.*
   (b) *$condition_q : E_q \longmapsto \mathbb{P}(\mathcal{V}_{Condition})$ places sets of conditions on edges of G_q.*
   (c) *$policy_q : E_q \longmapsto \mathbb{P}(\mathcal{V}_{Policy})$ labels edges in G_q with sets of policies.*
   (d) *$agreedBy_q : E_q \longmapsto \mathbb{P}(\bigcup_{q \in \mathcal{Q}} q.\mathsf{Agent.Name})$ maps an edge to a set of agents specified in the Agent attribute of the given NFR q. The agents are those who explicitly marked agreement with the valued metrics specified for the given edge (§3).*
   (e) *$setMetric_q : E_q \longmapsto \mathbb{P}(\mathcal{M} \times \mathcal{V}_{Metric})$ associates sets of valued metrics to edges of the timeline graph.*

Definition 2. *The **time domain** $(\mathbb{T}, \leq_t)$ is a set of **time points** $\mathbb{T}$ ordered by the binary relation $\leq_t$. As usual, $t_1 <_t t_2$ iff $t_1 \leq_t t_2$ and $t_1 \neq_t t_2$. The relation $\leq_t$ is linear, i.e., if $t_1 \leq_t t_2$ and $t_2 \leq_t t_1$ then $t_1 =_t t_2$ (antisymmetry), if $t_1 \leq_t t_2$ and $t_2 \leq_t t_3$ then $t_1 \leq_t t_3$ (transitivity), and $\forall t_1, t_2 \in \mathbb{T}, t_1 \leq_t t_2$ or $t_2 \leq_t t_1$ (totality—any two members of $\mathbb{T}$ are comparable under the relation $\leq_t$).*

Definition 3. *An **interval** I is a convex subset of $\mathbb{T}$.*

Notational conventions. Right endpoint of an interval I, denoted $r(I)$, is its supremum (also called its greatest upper bound); left endpoint, $l(I)$, of I is its infimum (least upper bound). $I = (t_i, t_j]$ is a left-open (i.e., $t_i \notin I$) and right-closed ($t_j \in I$) interval.

When applied, the chosen conceptualization gives a graph in which labels indicate (i) the conditions and policies provided and accepted by (ii) a group of stakeholders for setting (iii) specific target values for metrics that measure the degree of satisfaction of NFRs expected to be satisfied over the given interval. An important property of the timeline is that it is not linear: it branches, so that distinct edges can share same nodes. Timelines branch because of trade-offs between NFRs that cannot be satisfied simultaneously to ideal levels or because of change in conditions that require revision of target metric levels. However, knowing that target values must change with variation of target values of other metrics is helpful to a limited extent: in addition to variation, both the conditions under which variation is expected to occur and the policies to adopt in the aim of affecting the expected degree of the metrics are of interest. Consequently, each interval in the timeline is labeled with the Condition and Policy attributes.

Definition 4. *A **condition** describes constraints on the context or part thereof for the NFR (for which that timeline is being built), under which a given set of metric target values is considered desirable by a set of stakeholders.*

Knowing the conditions facilitates the identification of actions which will aim at changing or maintaining the given conditions. In other words, the content of Policy refers to the goals, processes, tasks, or actions whose achievement or execution leads, for the given interval, to the balacing of metric target values visible on the timeline for that interval.

Definition 5. *A **policy** is a goal, process, task, action, or a set thereof that the stakeholders believe needs to be achieved (for goals) or executed in the aim of ensuring the metric target values be reached under the condition associated to the policy in the given timeline.*

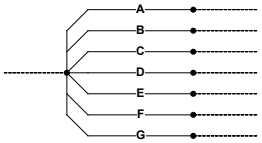
3 Using Timelines, Conditions and Policies

Information on time, conditions, and policies is closely related: analyzing one without the others is possible, but additional techniques can be applied if these considerations are treated together within a single cadre. Herein, two techniques, *defensive NFR planning* and *NFR planning* are presented. Each involves the acquisition of time and condition/policy information and its combination in order to arrive at a timeline for the given NFR. After timelines are acquired, they are then combined first locally within the same context (i.e., meaningfully related set of NFRs, such as the NFRs obtained by decomposing some higher-level NFR), and then accross contexts using in both cases the timeline synchronization technique (Technique 3). As the said technique generates a large number of alternative paths on synchronized timelines, pruning techniques are also suggested to assist in selecting the most desirable one on which the stakeholders can agree. The techniques presented in the remainder of the section thus integrate the various information acquired for the time and condition/policy considerations. Notice however, that timelines not only integrate this information, but also part of preferences information and record agreement between the stakeholders participating in the development process: preferences appear in the form of metrics and branches which result from alternative sets of metric values, themselves the result either of varying conditions, or disagreement of the stakeholders—those agreeing on labels for timeline edges are specified in the labeled timeline. Elicitation and analysis of preferences and agreement is relegated to later sections for clarity of presentation.

Technique 1. *(Defensive NFR planning.)* Defensive planning serves as a first step in making the software and the relevant part of its environment responsive to random events whose occurrence can affect the level of NFR satisfaction. Because of inherent unpredictability, the duration of such events and hence the starting and ending time are unlikely to be available. The construction of the timeline is condition-driven, whereby combinations of conditions are identified,

Airport capacity utilization efficiency

TIMELINE:



Label:	**A**	**B**	**C**	**D**	**E**	**F**	**G**
DURATION:	*varies*	*varies*	*varies*	*varies*	*varies*	*varies*	*varies*
CONDITION:	Normal conditions	Light weather contingency	Heavy weather contingency	Short term demand changes AND Normal conditions	Short term demand changes AND Light weather contingency	Short term demand changes AND Heavy weather contingency	Short term demand changes AND Heavy weather contingency
POLICY:	*No specific policy - usual operation*	Safety before capacity, Redirect to regional airports	Safety before capacity, Keep on ground, Redirect early to regional airports	Maintain average delay belows 35min	Safety before capacity, Redirect to regional airports	Safety before capacity, Keep on ground, Redirect early to regional airports	Safety before capacity, Keep on ground, Redirect early to regional airports, Avoid PopAreaZones
NFC & METRIC:							
Airport capacity utilization efficiency							
↑ **AirportCapUtil**	75%	55%	30%	68%	45%	30%	30%
Safety under weather contingency							
↓ **AirborneWaitingCont**		0.12	0.04		0.12	0.04	0.045
Flexibility under short-term changes in demand							
↓ **STChangesDelay**				2.15	2.27	2.27	2.27
↓ **SlotChangeDelay**				0.11	0.12	0.12	0.12
Environment noise pollution							
↓ **T&LOverPopAreaZones**	0%	5%	20%	5%	7%	30%	20%
AGREED BY	Airport representatives, Local city authorities	Weather Service Providers (WSP), Airport representatives, Local city authorities	Weather Service Providers (WSP), Airport representatives, Local city authorities	Airport representatives, Airline representatives, Local city authorities	Airport representatives, Airline representatives, Weather Service Providers (WSP), Local city authorities	Airport representatives, Airline representatives, Weather Service Providers (WSP)	Local city authorities

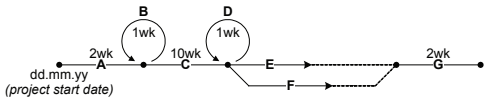
Fig. 1. Defensive NFR planning for the NFR Airport capacity utilization efficiency.

desired (target) metric values are given for all metrics of NFR that interact with the NFR for which the timeline is being built, stakeholders agreeing with the set metric values are identified and noted, alternative values are written down as well along with the stakeholders who suggested them, and finally, policies for meeting the given quality levels are referenced on the obtained timeline. In case an alternative is suggested, the condition line branches, as in Fig.1. In essence, timelines used for defensive NFR planning are decision trees which provide an overview of scenarios (i.e., sets of conditions) and associate with each a set of stakeholders, policies, and target metric values. A simple way to visualize such a timeline is with a table, as in Fig.1, where: conditions are noted above columns in which corresponding target metric values, agreed upon by a set of referenced stakeholders, are set; the bottom row identifies the policies the stakeholders suggested to meet the target quality levels.

Example 1. Fig.1 is an example of the output obtained by application of Technique 1. The graph indicates that all columns in the table are alternatives. Metric values are examples only. The metrics to list in the table are those that correlate with the metrics of the NFR for which the timeline is being built (the latter

Expenditure visibility

TIMELINE:



Label:	**A**	**B**	**C**	**D**	**E**	**F**	**G**
DURATION:	2 weeks	1 week	10 weeks	1 week	Until project end	Until project end	2 weeks
CONDITION:	Project start, Cost control team new to EMA	Unacceptably many errors in expenditure database	Error count acceptable in the EMA expenditure database	Significant discrepancies between forecast and actual exp.		Minor discrepancies between forecast and real expenditures	Project end announced, expenditure analysis and summary requested
POLICY:	Train Cost control team for quick exp. registration and early announc.	Train Cost control team to balance speed and accuracy	Train Cost control team to anticipate and forecast expenditures, and report these	Revise forecast, detect forecasting problems do more training for the Cost control team	Hire external expert to manage forecasting effort	Maintain recording, announc., and forecast performance	Stop recording, announc., and forecasting, produce analysis of all exp.
NFC & METRIC:							
Expenditure visibility							
↓ **TimeToRecExp**	8h	5h	3h	3h	2h	2h	
↑ **TimeTillExpAnn**	1 day	3 days	4 days	4 days	5 days	5 days	
Expenditure database accuracy							
↓ **ProbErroneousEntry**	High	High	Low	Low	Low	Low	
Expenditure forecast							
↓ **ProbErroneousForecast**			High	High	Low	Low	
Budget utilization							
↓ **ProbBudgetOverrun**	Med	Med	Med	Med	Low-Med	Low	
AGREED BY	PM, Accounting team, Cost control training team	PM, Accounting team, Cost control training team	PM, Accounting team, Cost control training team	PM, External forecast expert, Accounting team, Cost control training team	PM, Accounting team, Cost control training team, External forecast expert	PM, Accounting team, Cost control training team	PM, Accounting team, Cost control training team

Fig. 2. NFR planning with a timeline for the NFR Expenditure visibility.

are included in the table as well). Each column in the table contains the conditions, target metric values desired under the given conditions, the stakeholders who agree on these metric values, and the policies whose application is expected to lead to set metric levels. As discussed above, another visualization of the same information is a decision tree in which leaves indicate conditions, policies, stakeholders, and metric values, while branching occurs whenever conditions are combined. Without going into the detail of NFRs for the NFR listed in Fig.1, the metrics mentioned therein are specific to the ATM domain—for instance, AirportCapUtil compares the estimated or real capacity utilization with theoretical airport capacity, whereas T&LOverPopAreaZones may either compare levels of noise in noise sensitive areas (i.e., populated areas) over which takeoff and landing may be permitted under specific conditions, or the frequency of takeoff and landing over these areas. Notice that an arrow, oriented up for metrics to maximize, and down for those to minimize is also mentioned in the table. Observe further that combinations of conditions are indicated as for Short term demand changes, and that alternative sets of metrics, stakeholders, or policies are indicated by splitting the condition line, as for Heavy weather contingency when combined with Short term demand changes.

Technique 2. *(NFR planning.)* The NFR planning technique is applied instead of Defensive NFR planning (Technique 1) if: (i) conditions are predictable (which is often the case when defining quality levels for processes common in the given domain – here, e.g., communication during landing or takeoff, various security procedures, slot changes, and so on, as well as the project management processes associated to the development or evaluation project – see, Ex.2), and (ii) sequence of conditions may matter (in Defensive nfr planning, sequence is difficult to establish because of unpredictability). The graph representation of the timeline tends then to differ in NFR planning in that the timeline does not involve alternatives only (or mostly) as is the case in Defensive NFR planning. To simplify the reading of the timeline, loops indicate edges that can be optionally taken, whereby it is assumed that the loop is executed only once if no other indication is given in the graph (such as the maximal number of times the loop can be executed).

Example 2. Fig.2 shows a tabular representation of timeline information and gives a graphical representation of the timeline to show alternatives in the form of branches and loops. Intervals in the timeline carry alphabetical labels to associate the two parts of the visualization.

Technique 3. *(Timeline synchronization.)* Synchronization amounts to combining timelines so as to build the general NFR timeline which integrates conditions and policies associated with all of the individual NRF timelines. Accomplishing this manually while accounting for the various possible alternatives is inappropriate, as the merging of two relatively simple timelines illustrates: in Fig.3, the usual representation with loops is first expanded for both timelines to show the alternative paths, while the rightmost part of the figure shows a part of the combined timeline in which many alternatives are represented. Synchronization is thus partly automated to make the task feasible; it proceeds as follows:

1. Select two timelines T_{q_1} and T_{q_2} in the same context.
2. Apply the **GenerateAlternatives** algorithm (see below) to $\mathcal{T}_{q_1}$ and $\mathcal{T}_{q_2}$ generate a third timeline $\mathcal{T}_{q_3}$ which contains all possible alternative paths obtained by combining the intervals from each of the two timelines.
3. Apply pruning techniques (see below) to reduce the number of relevant alternative paths in $\mathcal{T}_{q_3}$.

Example 3. Fig.3 shows the partial result of synchronizing two timelines. Following the notation established earlier (see, Fig.2), each label on the interval is associated with conditions, target metric values, stakeholders, policies, and durations. Combining these results in intervals which carry, at first simply combined labels, but later require checking for consistency—the principal indicator for checking and possible revision being the negative interactions (obtained from correlations) between metrics whose trade-offs are not accounted for in individual timelines, but do need to be treated in the synchronized timeline (i.e., a metric in $\mathcal{T}_{q_1}$ interacts negatively with a metric in $\mathcal{T}_{q_2}$). Observe in the middle of Fig.3 that the time is represented above the extended timeline forms so that the length of the intervals be proportional to their durations.

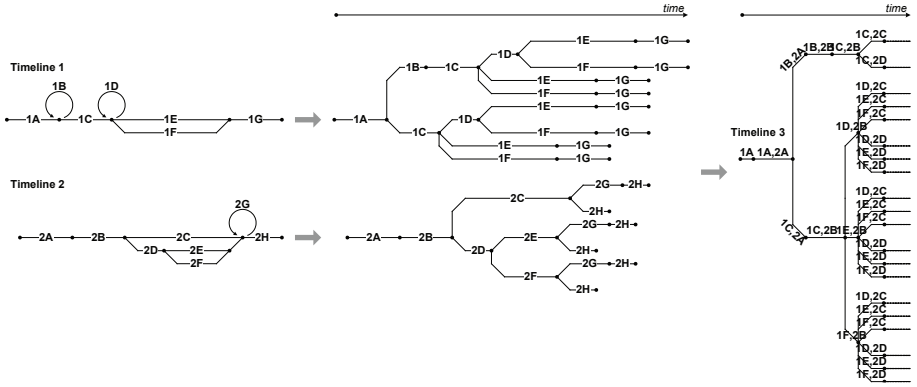


Fig. 3. Synchronizing two arbitrary timelines.

Technique 4. (GenerateAlternatives algorithm) The algorithm takes two timelines $\mathcal{T}_{q_1}$ and $\mathcal{T}_{q_2}$ to produce a third timeline $\mathcal{T}_{q_3}$ which contains all allowed alternative paths obtained by combining $\mathcal{T}_{q_1}$ and $\mathcal{T}_{q_2}$. First, the set of timepoints is built for $\mathcal{T}_{q_3}$. Let

$\mathcal{T}_{q_1} = (G_{q_1}, F_{q_1})$,
$\mathcal{T}_{q_2} = (G_{q_2}, F_{q_2})$,
$T_{q_1} = \{t_i \mid \forall n_i \in N_{q_1}, t_i = timing_{q_1}(n_i)\}$, and
$T_{q_2} = \{t_i \mid \forall n_i \in N_{q_2}, t_i = timing_{q_2}(n_i)\}$.

The first part of the algorithm generates the set of timed nodes T_{q_3} for $\mathcal{T}_{q_3}$, provided a complete ordering of nodes in $\mathcal{T}_{q_1}$ and $\mathcal{T}_{q_2}$ is available (i.e., it is enough to associate arbitrary times to nodes, while only ensuring their order corresponds to what is expected by the stakeholders):

```
01  begin GenerateAlternatives
02  i ← 0; T_q3 ← ∅; N_temp ← N_q1 ∪ N_q2
03  while  t_i ≤ Max(T_temp) and t_i ∈ T_temp do
04         t_i ← Min(T_temp)
05         if t_i ≠ t_i−1 then T_q3 ← T_q3 ∪ {t_i} end if
06         T_temp ← T_temp \ {t_i}; i ← i + 1
07  end while
```

Paths from $\mathcal{T}_{q_1}$ and $\mathcal{T}_{q_2}$ are then combined to construct P_{q_3} ($p^i_{q_3}$ denotes the i-th path in timeline $\mathcal{T}_{q_3}$). It is assumed that a function $getPaths : \{\mathcal{T}\} \longrightarrow \{P\}$ returns the set of all paths of a timeline:

```
08  i ← 0; j ← 0; k ← 0; P_q1 ← getPaths(T_q1); P_q2 ← getPaths(T_q2)
09  while k ≤ |P_q1| do
10    while j ≤ |P_q2| do
11      while t_i ≤ Max(T_q3) do
12        p^i_q3 ← p^i_q3 ∪ {E_i : l(E_i) = Min(t_i ∈ p^k_q1, t_i ∈ p^j_q2) and
```

$r(E_i) = Min(\{t_i, t_{i+1} \in p^k_{q_1}\} \cup \{t_i, t_{i+1} \in p^j_{q_2}\} \setminus \{l(E_i)\})$

$E' \leftarrow E \in E_{q_1}, E \subseteq E'; E'' \leftarrow E \in E_{q_2}, E \subseteq E''$

$setMetric_{q_3}(E_i) \leftarrow (setMetric_{q_1}(E') \cup setMetric_{q_2}(E''))$

$agreedBy_{q_3}(E_i) \leftarrow (agreedBy_{q_1}(E') \cup agreedBy_{q_2}(E''))$

$condition_{q_3}(E_i) \leftarrow (condition_{q_1}(E') \cup condition_{q_2}(E''))$

$policy_{q_3}(E_i) \leftarrow (policy_{q_1}(E') \cup policy_{q_2}(E''))$

$P_{q_3} \leftarrow P_{q_3} \cup p^i_{q_3}; i \leftarrow i + 1$

end while

$j \leftarrow j + 1$

end while

$k \leftarrow k + 1$

end while

Let *pathsToEdges* be the function that takes the set of paths and returns the unique set of edges employed to build the given paths. The third timeline is then defined as follows:

$E_{q_3} \leftarrow pathsToEdges(P_{q_3})$

$F_{q_3} \leftarrow \{setMetric_{q_3}, agreedBy_{q_3}, condition_{q_3}, policy_{q_3}, timing_{q_3}\}$

$\mathcal{T}_{q_3} \leftarrow (T_{q_3}, E_{q_3}, F_{q_3})$

end GenerateAlternatives

Example 4. Fig.3 illustrates the use of the algorithm by visualizing the two input timelines and the result as a branching timeline, easily obtained from the formal timeline definitions.

By progressively combining all NFRs' timelines, the general NFR timeline can be constructed. As visible from the algorithm, visual check of the obtained timeline is required to ensure combined conditions, policies, and metric values are not inconsistent. If inconsistencies are detected (taking the form of, e.g., incompatible metric values or preferences, contradictory conditions or policies), they ought to be resolved by either adjusting the problematic paths or eliminating them. Ideally, these checks should be performed early, after each application of the algorithm so that problems are detected and resolved before moving further with timeline synchronization. It is apparent that combining timelines using the GenerateAlternatives algorithm results in complex trees difficult to analyze visually for more than a few NFRs. The following technique are used to reduce the set of alternative paths to consider.

Technique 5. (Pruning criterion: Pareto efficiency) Under the reasonable assumption of stakeholder rationality, and knowing that changing the metric target values may affect, if interaction is present, other metrics' target values, any path to maintain in the timeline must be Pareto efficient.[1] In the present terminology, a path is Pareto efficient if there is no interval on it such that a target metric value associated to this interval can be improved without degrading the target

[1] In economics, an allocation of resources between economic agents is Pareto efficient if any change in the allocation intended to improve the part of some agents makes at least some other agents worse off.

value of another metric on the same path. That degradation will occur is known from the correlation between metrics.

Other criteria and associated techniques for path selection not discussed in the present paper, though relevant are:

- *Pruning by probable paths*: If statistical data is available on the ocurrence of conditions and the execution of processes fulfilling the policies indicated in the timeline (data may come from, e.g., comparable processes put in place in other project), probabilities can be associated with conditions and policies in the aim of selecting the most probable paths.
- *Pruning by risk*: Identifying least preferred paths may be of interest (this can be driven by identification of least-preferred intervals, which is simpler to execute than by observing entire paths), as the conditions associated to them clearly pose a risk in terms of quality levels. Knowledge of problematic conditions may provide indications useful for revising conditions and policies on preferred paths in order to make them more robust. Such revision can be executed by applying Defensive NFR planning over preferred paths.
- *Priority-based highest-metric pruning*: A classification of paths by aggregate priority level and aggregate metric target level may point automatically to prioritized and most preferred paths to consider.

As no single technique guarantees by itself an appropriate choice of path, combining several of the above techniques may provide insight from different perspectives and reduce the set of interesting paths to a manageable number.

4 Conclusions

This paper opens up a discussion of the specification and analysis of time properties of NFRs. The aim has been to suggest basic useful concepts and analyses and point to difficulties. More thorough study of both the specification and analysis of NFR temporal properties is necessary, as the variation of NFR satisfaction levels over time is a fact. Various temporal formalisms are available, and their relative worth should now be studied in light of the likely (un)availability of information. Methodological considerations will receive considerable attention, given the difficulty to predict and plan for the variation of NFR satisfaction levels. Our current focus is on additional analysis techniques, the integration of the concepts discussed herein with NFR, and the automation of the techniques.

References

1. Eurocontrol. *ATM User Requirements Document (Vol. 1 & Vol. 2)*. European Air Traffic Control Harmonisation and Integration Programme, Ref FCO.ET1.ST04.DEL01, European Org. for the Safety of Air Navigation, 1999.
2. J. Mylopoulos, L. Chung, B. Nixon. Representing and Using Nonfunctional Requirements: A Process-Oriented Approach. *IEEE Trans. Softw. Eng.* 18(6):483–497, 1992.
3. L. Osterweil. Strategic Directions in Software Quality. *ACM Computing Surveys*, 28(4):738–750, 1996.