Achieving, Satisficing, and Excelling

Ivan J. Jureta[1], Stéphane Faulkner[1], and Pierre-Yves Schobbens[2]

[1] Information Management Research Unit, University of Namur, Belgium
[2] Institut d'Informatique, University of Namur, Belgium
iju@info.fundp.ac.be, stephane.faulkner@fundp.ac.be, pys@info.fundp.ac.be

Abstract. Definitions of the concepts derived from the goal concept (including functional and nonfunctional goal, hardgoal, and softgoal) used in requirements engineering are discussed, and precise (and, when appropriate, mathematical) definitions are suggested. The concept of satisficing, associated to softgoals is revisited. A softgoal is satisfied when thresholds of some precise criteria are reached. Satisficing does not cover situations in which continual improvement of thresholds is expected. The notion of *excelling* is suggested to cover such cases, along with the concept of *disposition* to represent and reason about excelling.

1 Outline

One motive for representing and reasoning about a system is to precisely understand the purpose thereof and subsequently use this information to identify, analyze, and select among alternative properties and behaviors needed of the system to fulfill its purpose. The *goal* concept stands out among the various abstractions proposed for the representation and reasoning about a system's purpose. It is now accepted that the concept is relevant for the elicitation, elaboration, structuring, specification, analysis, negotiation, documentation, and modification of stakeholders' requirements on a system [1,2,3,4,5,6,7,8,9,10,11,12,13].

Few contributors to the field of requirements engineering (RE) agree on a *precise* definition of the goal concept. Elasticity in definitions may facilitate the basic understanding of goal-based RE frameworks to non-experts: common knowledge substitutes for specialized RE knowledge, thus facilitating learning. Elasticity, however, also involves difficulties in communication, imprecision in intended meaning, and overuse and/or abuse of the terminology.

In response, we study definitions of the goal and its derived concepts, including hardgoal, softgoal, functional goal, and nonfunctional goal. We suggest precise definitions consistent with the literature; when appropriate, definitions are in formal logic. It is usually said that a hardgoal can be achieved, while a softgoal can only be satisficed. We revisit the concept of satisficing, commonly associated to the concept of softgoal. A softgoal is satisfied when thresholds of some precise criteria are reached. We argue that satisficing does not cover situations in which continual improvement of thresholds is preferred. We subsequently suggest the notion of *excelling* to cover such cases, along with the concept of *disposition* to represent and reason about excelling. The contributions of this paper are: (i)

the set of precise definitions of goal and derived concepts of hardgoal, softgoal, functional goal, and nonfunctional goal; (ii) the notion of excelling intended to complement the concepts of achievement and satisficing in goal-oriented RE; and (iii) the concept of disposition intended for representing and reasoning about requirements to which the notion of excelling applies.

2   Goal Concept in RE Research

System development frameworks include, since the 1970s some form of analysis involving goals [9], among them context analysis, definition study, and participative analysis. Goals have become an essential part of any system's documentation through standards such as e.g., the IEEE-Std-830/1993. There is no established definition for the goal concept: consider Table 1, which lists informal definitions of the goal concept appearing in various goal-oriented RE frameworks. KAOS highlights the nonoperational nature of goals, pointing to the need for taking action to make goals precise by refinement (see, e.g., [2]). Broadly speaking, the KAOS definition is in line with those of Tropos, i^*, GDC, and Lightswitch: a goal designates desirable conditions on the system and/or its environment. Such condititions restrict the set of alternative system and environment states, so that it is appropriate to say that a goal describes desired states. A different conceptualization appears in NFR, where goals are employed for representing nonfunctional requirements, in addition to design decisions, and arguments for or against other goals. We can interpret "design decisions" as restricting potential desired system and environment states. Notions of argument and justification appear in NFR and GBRAM. We have discussed elsewhere [14] the relevance of argumentation and justification for goal-oriented RE, arguing and illustrating that it is more appropriate to maintain the notion of argument separate from the goal concept.

Regarding the use of goals for modeling nonfunctional requirements, two relevant goal taxonomies have been introduced since the seminal contributions in the NFR framework (see, e.g., [9,18] for discussions).[1] *Functional* goals are distinguished from *nonfunctional* ones, and *softgoals* from *hard goals. Functional* goals have been used to represent services that the software is expected to deliver (i.e., *what* the software does), whereas *nonfunctional* goals refer to quality requirements that the software needs to satisfy while delivering the services (i.e., *how* the software provides services; e.g., securely, safely, rapidly, etc.). While it is common to equate nonfunctional goals and softgoals (e.g., [1]), it has been subsequently argued that *softgoals* belong to another taxonomy, in which they are distinguished from *hardgoals* [9]. According to the traditional definition, "a softgoal is similar to a (hard) goal except that the criteria for whether a softgoal is achieved are not clear-cut and a priori." [19] The definition used in the REF framework [15] adds details, as shown in Table 1.

While softgoal satisfaction cannot be established in a clear-cut sense [1], the satisfaction of a hardgoal is objective in that it can be established using (formal) verification techniques [2]. In this respect, a hardgoal is said to be *achievable*,

---

[1] This paragraph follows our previous discussions on the subject [18].

Table 1. Informal definitions of the *goal* concept in goal-oriented RE

Framework	Informal definition of the goal (and derived) concepts
KAOS	"A goal is a nonoperational objective to be achieved by the composite system. Nonoperational means that the objective is not formulated in terms of objects and actions available to some agent in the system; in other words, a goal as it is formulated cannot be established through appropriate state transitions under control of one of the agents." [2] "A goal is a desired property about quantities in the environment." [8]
Tropos and i^*	"A goal is a condition or state of affairs in the world that the stakeholders would like to achieve." [3,5,10]
NFR	"Goals [represent] non-functional requirements, design decisions and arguments in support or against other goals." [1,7]
REF	"According to the nature of a goal, a distinction is made between hard goals and soft goals. A goal is classified as hard when its achievement criterion is sharply defined [...]. For a soft goal, instead, it is up to the goal originator, or to an agreement between the involved agents, to decide when the goal is considered to have been achieved [...]. In comparison to hard goals, soft goals can be highly subjective and strictly related to a particular context; they enable the analysts to highlight quality issues [...] from the outset [...]" [15]
GDC	"An enterprise goal is a desired state of affairs that needs to be attained." [16]
GBRAM	"Goals are high level objectives of the business, organization or system. They capture the reasons why a system is needed and guide decisions at various levels within the enterprise." [4]
Lightswitch	"[A] maintenance goal is said to represent a condition that remains constant. [...] [An] achievement goal has definite pre and post-conditions. The pre-condition represents the interpretation that the state of affairs has drifted (or will drift) outside of the threshold associated with the norm [i.e., a variable of the system whose state the system attempts to maintain unchanged as defined by an observer]. The post condition is an interpretation that is within this threshold." [17]

whereas a softgoal is *satisficeable* [1,7,12]. The concept of *satisficing* originates in H. Simon's work [20] in economics: to satisfice is to set a threshold, and accept any achievement above the threshold. In addition to involving satisficing, a softgoal has a subjective component, in that various stakeholders of the system will have different thresholds. We have worked on a more expressive definition of softgoals elsewhere [18], but we did not provide a mathematical definition.

The KAOS framework provides the most precise hardgoal conceptualization: a hardgoal is defined in terms of predicate patterns in a discrete linear temporal first-order logic (see, e.g., [21]).[2] A hardgoal is any one of the following [8]: an

---

[2] In publications on KAOS, what we call hardgoal here is called simply "goal". Note, however, that this conceptualization does not encompass the softgoal concept (which was introduced separately from KAOS): if we know a constraint, written in logic over system histories, we can check at any time if the actual history of the system respects or not the constraint.

achieve hardgoal (pattern: $\phi \Rightarrow \diamond\psi$), a cease hardgoal ($\phi \Rightarrow \diamond\neg\psi$), a maintain hardgoal ($\phi \Rightarrow \square\psi$), an avoid hardgoal ($\phi \Rightarrow \square\neg\psi$). The same conceptualization is adopted in Formal Tropos [11], where patterns are used in the same way to define hardgoals. An informal interpretation of the said conceptualization is that a hardgoal is a constraint over system histories (i.e., behavior over time).

It is clear that the goal concept is intended to be rich in meaning. Instead then of seeking an all encompassing definition, we study derived concepts, obtained by crossing the hardgoal/softgoal and functional/nonfunctional taxonomies; we thus have: (i) *functional hardgoals*, which are hardgoals about what the system should do (e.g., in an email application, such a goal can be: "whenever an e-mail marked as important arrives, the user is informed with a pop-up window and a sound"); (ii) *nonfunctional hardgoals* which describe verifiable criteria for how the system should operate (e.g., "the user should be informed about important e-mail arrival within 1 second of arrival"); (iii) *functional softgoals* describe a subjective requirement of a stakeholder about what the system should do (e.g., "the user should be informed when an e-mail marked as important arrives"); and (iv) *nonfunctional softgoals* which indicate in a subjective and nonverifiable manner how the system should operate (e.g., "the user should be informed rapidly about the arrival of an e-mail marked as important").

In summary, there is no unique definition of goal. One reason for this is that the goal concept is intended to be rich in meaning. Variations in definitions are also due to slightly different uses of the concept in each framework. Whether a goal conceptualization is appropriate depends on how useful is the framework in which it is used. A prescriptive general definition thus seems excluded. It remains, however, of interest to seek a conceptual framework in which the derived concepts mentioned above can be used together, so that the benefits of these complementary concepts can be combined when representing and reasoning about requirements. A precise definition is already available for the hardgoal concept. We can now suggest a common ground for the cited derived concepts.

3   A Common Framework

Consider a toy system that has only two properties, p_1 and p_2. All possible combinations of allowed values for p_1 and p_2 define all possible states of the system. Let S_1, S_2, and S_3 be arbitrary system states, as shown in the bottom part of Figure 1. Assume that measurements are performed on the system in order to evaluate its quality. To perform measurement, we define two metrics d_1 and d_2. To relate what we observe in the system and the values of the metrics, we define mappings M_1, M_2, and M_3 between system states and value combinations of the two metrics. Since some minimum level of quality is expected, we define thresholds on metrics: in Figure 1, $t_1 \equiv d_1 \geq d_1^t$ and $t_2 \equiv d_2 \geq d_2^t$, so that the quality is above the minimal level only when the system is in state S_2 and not in the other two.

Taking the state-based conceptualization of the hardgoal concept, we define a hardgoal $hg_1 \equiv (\top \Rightarrow \square(p_2 = p_2^*))$ as a value p_2^* of the property p_2. hg_1 is
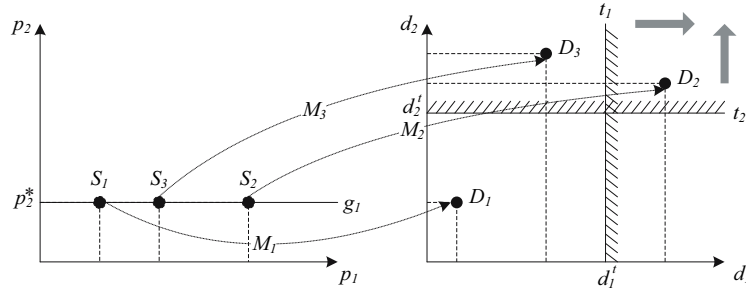
Fig. 1. Arbitrary functional and nonfunctional hardgoals in a toy system

a functional hardgoal, since it says precisely what the system is expected to do (i.e., set property p_2 to value p_2^*). Returning to the informal understanding of the nonfunctional hardgoal given earlier (§2), we see that it cannot be defined over system properties, but on metrics defined for the system. We can thus define two nonfunctional hardgoals in Figure 1: $\tilde{hg}_1 \equiv (d_1 \geq d_1^t)$ and $\tilde{hg}_2 \equiv (d_2 \geq d_2^t)$.

Are there any softgoals in Figure 1? We know that a softgoal is used to model requirements at the earliest stages of an RE process (e.g., [1,9,10,12,18]). Usually, initial requirements, and consequently softgoals are imprecise, subjective, idealistic, and context-specific [18], meaning that we cannot have a softgoal in Figure 1—the figure is already too precise. Consequently, and in line with contributions in the RE field, a softgoal is here understood as an initial, early form of requirements about what the system should do and how "well" it should do it, from which one or more functional and/or nonfunctional hardgoals are extracted during the requirements process. This is appropriate, since we also know that one cannot manage (i.e., assure, control, or improve) what one cannot measure (e.g., [22,23,24,25,26]): if quality-related information contained in softgoals is to be used in decision-making during an RE process, we need to make a softgoal precise, agreed upon by various stakeholders, and realistic—that is, we need to convert it into nonfunctional hardgoals. Same applies for functional softgoals: we require precise, agreed upon, and realistic requirements about what services the system should deliver in order to be able to implement them in later stages of the system development process.

Having established that softgoals appear earlier in an RE process than hardgoals, recall that softgoals are associated to the concept of satisficing. Satisficing is the reason we specified our nonfunctional hardgoals as thresholds only, instead of, e.g., more elaborately stating the desired values of d_1 and d_2. Indeed, nonfunctional hardgoals derived from nonfunctional softgoals serve in RE as criteria for comparing alternative system structures (e.g., [1,7,9,10]). A system structure is chosen over an alternative one if the former dominates the latter over a set of nonfunctional softgoals or the derived nonfunctional hardgoals. In our toy system, we would choose a system structure that is associated to higher values of the two metrics, over one associated with lower values; we would discard structures that are not above both thresholds. Satisficing, while clearly useful

and reflecting the inability to identify *the* optimal system structure, does not cover requirements in which continuous improvement is sought. Indeed, satisficing does not go as far as to say what values above a threshold are preferred over other values, also above the threshold. That is, all values are equally desired, provided that they are above the threshold. In many actual cases, we do need to set thresholds, but we need not equally prefer all above-threshold structures. This is the case in particular for adaptable systems based on the agent or services paradigms. We encountered this need in an actual setting: we proposed elsewhere [27] an adaptable system in which above-threshold structures are learned. Therein, a "system structure" corresponds to a composition of web services that allows a service request (coming from a system user and specified in terms of requirements) to be filfiled. To form compositions, a composer web service observes other web services during execution and subsequently selects (for participation in a composition) only those that allow it to obtain more desired values over a given set of metrics. Compositions are revised, so that the quality to which same service requests are fulfilled improves over time. When specifying requirements on such a system, we are clearly not interested only in satisficing—if we did rely on satisficing only, we would not exploit the ability of the system to improve the fulfillment of service requets. We would not exploit the system's ability to adapt. Instead, we need to express that the system both needs to satisfice (so that below-threshold compositions be discarded) and to always improve compositions. We clearly cannot use the notion of satisficing to express requirements on continuous improvement: instead, we use the notion of *excelling* to do so. The limitation of satisficing that we highlight here is not novel: recently, J. L. Pollock proposed the concept of locally global planning, in which "any plan with a positive expected utility is defeasibly acceptable, but only defeasibly. If a better plan is discovered, it should supplant the original one. Satisficing would have us remain content with the original." [28] This discussion brings us to the following position: to use the concept of softgoal grounded in satisficing to express requirements that are associated to the concept of excelling is to extend the softgoal concept too far. We thus propose the concept of *disposition*. A disposition is a preference order defined over goals of the same type; we thus have the following taxonomy for the disposition concept: (i) *hard-functional disposition*, defined over functional hardgoals; (ii) *hard-nonfunctional disposition*, over nonfunctional hardgoals; (iii) *soft-functional disposition*, over functional softgoals; and (iv) *soft-nonfunctional disposition*, over nonfunctional softgoals. An example of a hard-nonfunctional disposition expressed informally is: "the user should be informed about important e-mail arrival within the least time possible" generalizes a preference order in which it is clear that the nonfunctional hardgoal "the user should be informed about important e-mail arrival within 0.5 second of arrival" is preferred to "the user should be informed about important e-mail arrival within 1 second of arrival". Note the following:

– Do not mistake excelling for optimization: the latter applies if the email system is designed so that it always gives the optimal time (i.e., 0 seconds). This is clearly idealistic. Excelling is in a sense optimization over time and

given resource boundedness; that is, the email system excels if it reduces time for informing the user at each email arrival compared to the time it needed on the last occasion an email arrived. Excelling applies even if the system does not continually improve (expecting this may be idealistic); what is important for excelling to apply is that, even if, in our example email system, notification time increases, it restablishes and goes down at some later and observable point (i.e., not indefinetly in the future).

– Not always can be a disposition so summarily expressed as in our notification time example for the email application. It may for instance happen that disjoint subsets of metric values are preferred, so that a disposition does not reduce to a formulation of desired direction for metric values. We explore in the remainder a simple notion of disposition mainly because we are introducing the concept here—extensions to its expressivity are of interest in current and future effort.

To express our various types of goals, we start from the multi-sorted first-order version of MITL [29,30], a continuous real-time linear temporal logic. Some predefined sorts (e.g. real numbers) have a fixed interpretation that will be used to express metrics. Our logic starts from Φ, a first-order vocabulary, consisting of predicate and function symbols p, f. They can be declared as flexible (time-dependent) or rigid. As usual, constant symbols are viewed as 0-ary rigid function symbols. Starting with atomic formulas of first-order logic, we form more complex formulas as usual by closing off under truth-functional connectives (i.e., $\land$, $\lor$, $\neg$, and $\rightarrow$)[3], temporal operators (i.e., next $\bigcirc$, eventually $\diamond$, always $\square$, until U and unless W) that can be indexed by a non-singular real-time constraint, existential ($\exists$) and universal ($\forall$) quantification. We denote the resulting language $\mathcal{L}$. We interpret formulas of $\mathcal{L}$ over the structure $\mathcal{T} \equiv \langle \mathcal{D}_\mathcal{S}, \mathcal{S}, \pi, \mathcal{H} \rangle$, where $\mathcal{D}$ gives a domain to interpret each sort, $\mathcal{S}$ is a set of states of the system, π is an interpretation assigning each predicate symbol and function symbol in Φ a predicate or function of the right arity over $\mathcal{D}$, if the symbol is declared rigid. If the symbol is declared flexible, it depends furthermore on the current state. $\mathcal{H}$ is a set of timed state sequences, $\mathcal{S}_0, I_0, \mathcal{S}_1, I_1 \ldots$ i.e. an infinite sequence of *states* and their associated interval of time, representig all possible executions of the system. These intervals I_i must partition the positive reals. To interpret first-order variables, we use a valuation function σ, which, given a variable of sort s returns its value, an element of $\mathcal{D}_f$. Given a structure $\mathcal{T}$, an history h, a time t and a valuation σ, we can associate with every formula of $\mathcal{L}$ a truth value in the usual way. A formula holds in a structure if it yields true for all histories, valuations and times. We call the obtained logic $\mathbb{L}_\mathcal{L}$. We can now give a precise definition of functional hardgoal.

Definition 1. *A functional hardgoal is a formula in $\mathbb{L}_\mathcal{L}$ that restricts the possible histories of a given system only to those desired by system stakeholders.*

To express the nonfunctional hardgoal concept, we use *metrics*. A metric is a rigid function symbol, which will return values in a sort equipped with an order.

---

[3] Usual abbreviations apply, e.g., $(\phi \Rightarrow \psi) \equiv \square(\phi \rightarrow \psi)$.

Definition 2. *A nonfunctional hardgoal is a formula in $\mathbb{L}_\mathcal{C}$ that restricts the values of metrics to those desired by system stakeholders.*

To relate the metrics and the behaviour of the system (recall Figure 1), we also need mappings between functional and nonfunctional hardgoals.

Definition 3. *A hardgoal mapping is a formula in $\mathbb{L}_\mathcal{C}$ over one or more functional hardgoals and one or more nonfunctional hardgoals.*

Taking the email application example, the following is a functional hardgoal, and is followed by an equivalent nonfunctional hardgoal:

$hg \equiv [\forall m : Email \; (arrived(m) \wedge important(m)) \Rightarrow$
$\Diamond_{\leq 1sec} \exists \; w : PopupWindow, s : NotifSound \; (display(w) \wedge play(s))]$

$\tilde{hg} \equiv [timeToNotification \leq 1sec]$

We can then define a hardgoal mapping for the above as follows: $hg \equiv \tilde{hg}$.

In contrast, we understand softgoals as expressing dispositions, i.e. preference over goals of the same type. To accommodate dispositions, we extend $\mathbb{L}_\mathcal{C}$ in the following way. First, we add a new kind of formulas, *disposition formulas*, such that if ϕ and ψ are formulas of $\mathcal{L}$ then $\phi \succeq_d \psi$ is a disposition formula. We take here the simplest case, in which we do not allow the operator $\succeq_d$ to appear in, e.g., temporal formulas of the language. $\mathcal{L}$ extended with disposition formulas is denoted $\mathcal{L}^{\succeq_d}$. Second, we extend our structures $\mathcal{T}$ to interpret defeasible formulas: we add a function υ which maps a real number (informally understood as a utility value) to non-disposition formulae. υ is then evaluated as follows: if $\phi \succeq_d \psi$, then $\upsilon(\phi) \geq \upsilon(\psi)$, which means that ϕ is preferred to ψ.

Following the earlier example, we may have the following hard-nonfunctional disposition:

$[timeToNotification \leq 0.5sec] \succeq_d [timeToNotification \leq 1sec]$

We define a disposition in terms of hard and soft disposition as follows:

Definition 4. *A hard disposition is a disposition formula in $\mathbb{L}_{\mathcal{L}^{\succeq_d}}$ between hardgoals.*

Definition 5. *A soft disposition is a preference either between only functional softgoals or between only nonfunctional softgoals.*

4   Conclusions and Future Work

Definitions of the concepts derived from the goal concept (including functional and nonfunctional goal, hardgoal, and softgoal) used in RE are discussed, and precise (and, when appropriate, mathematical) definitions are suggested. The concept of satisficing, associated to softgoals is revisited. A softgoal is satisficed when thresholds of some precise criteria are reached. Satisficing does not cover

situations in which continual improvement of thresholds is expected. The notion of *excelling* is suggested to cover such cases, along with the concept of *disposition* to represent and reason about excelling.

Although we have only presented here the simplest notion of disposition, we believe that the paper opens a particularly relevant discussion for goal-oriented RE. We hope it motivates similar efforts to ours in exploring more expressive concepts and techniques for RE. More elaborate expressions of dispositions need to be possible if excelling is to be properly accounted for; these include, e.g., conditional dispositions. We are working on extending the expressivity of the concept and are building a method to use the disposition concept in a systematic manner during the RE process. The method is intended to extend established goal-oriented RE frameworks. Tool support will be explored.

References

1. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Trans. Softw. Eng. 18(6), 483–497 (1992)
2. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20 (1993)
3. Yu, E.: Modeling Strategic Relationships for Process Reengineering. PhD thesis, Dept. of Computer Science, University of Toronto (1994)
4. Anton, A.I.: Goal-based requirements analysis. In: Proceedings of the International Conference on Requirements Engineering (1996)
5. Yu, E.: Towards modeling and reasoning support for early requirements engineering. In: Proceedings of the IEEE International Symposium on Requirements Engineering (1997)
6. Roland, C., Souveyet, C., Achour, C.B.: Guiding goal modeling using scenarios. IEEE Transactions on Software Engineering, Special Issue on Scenario Management, 1055–1071 (1998)
7. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Dordrecht (1999)
8. Letier, E.: Reasoning about Agents in Goal-Oriented Requirements Engineering. PhD thesis, Dept. d'Ingenierie Informatique, Universite de Louvain (2001)
9. van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: RE 2001. Proceedings of the International Symposium on Requirements Engineering (2001)
10. Castro, J., Kolp, M., Mylopoulos, J.: Towards requirements-driven information systems engineering: the tropos project. Information Systems 27(6), 365–389 (2002)
11. Fuxman, A., Liu, L., Mylopoulos, J., Pistore, M., Roveri, M., Traverso, P.: Specifying and analyzing early requirements in tropos. Requirements Engineering 9(2), 132–150 (2004)
12. Letier, E., van Lamsweerde, A.: Reasoning about partial goal satisfaction for requirements and design engineering. In: Proceedings of the International Conference on Foundations of Software Engineering (2004)
13. van Lamsweerde, A.: Goal-oriented requirements engineering: A roundtrip from research to practice. In: RE 2004. Proceedings of the International Requirements Engineering Conference (2004)

14. Jureta, I.J., Faulkner, S., Schobbens, P.Y.: Justifying goal models. In: RE 2006. Proceedings of the 14th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, Los Alamitos (2006)
15. Donzelli, P.: A goal-driven and agent-based requirements engineering framework. Requirements Engineering 9, 16–39 (2004)
16. Kavakli, E.: Goal-Driven Requirements Engineering: Modeling and Guidance. PhD thesis, University of Manchester (1999)
17. Regev, G., Wegeman, A.: Where do goals come from: the underlying principles of goal-oriented requirements engineering. In: RE 2005. Proceedings of the International Requirements Enginering Conference (2005)
18. Jureta, I.J., Faulkner, S., Schobbens, P.Y.: A more expressive softgoal conceptualization for quality requirements analysis. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, Springer, Heidelberg (2006)
19. Liu, L., Yu, E.: Designing information systems in social context: a goal and scenario modeling approach. Information Systems 29, 187–203 (2004)
20. Simon, H.: A behavioral model of rational choice. Quarterly Journal of Economics 59, 99–118 (1955)
21. Manna, Z., Pnuelli, A.: The Temporal Logic of Reactive and Concurrent Systems. Springer, Heidelberg (1992)
22. IO for Standardization ISO 8402 Quality management and quality assurance - Vocabulary. International Organization for Standardization (1986)
23. Kitchenham, B., Pfleeger, S.L.: Software quality: The elusive target. IEEE Softw. 13(1), 12–21 (1996)
24. Briand, L.C., Morasca, S., Basili, V.R.: An operational process for goal-driven definition of measures. IEEE Trans. Softw. Eng. 28(12), 1106–1125 (2002)
25. Fenton, N.E., Neil, M.: Software metrics: roadmap. In: ICSE - Future of SE Track, pp. 357–370 (2000)
26. Haag, S., Raja, M., Schkade, L.: Quality function deployment usage in software development. Communications of the ACM 39(1), 41–49 (1996)
27. Jureta, I.J., Faulkner, S., Achbany, Y., Saerens, M.: Dynamic web service composition within a service-oriented architecture. In: ICWS 2007. Proceedings of the International Conference on Web Services (2007)
28. Pollock, J.L.: Thinking about Acting: Logical Foundations for Rational Decision Making. Oxford University Press (Forthcoming)
29. Alur, R., Feder, T., Henzinger, T.: The benefits of relaxing punctuality. In: Proceedings of the Tenth Annual Symposium on Principles of Distributed Computing, pp. 139–152. ACM Press, New York (1991)
30. Henzinger, T., Raskin, J.F., Schobbens, P.Y.: The regular real-time languages. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 580–591. Springer, Heidelberg (1998)