Quantitative Modal Transition Systems
(Invited Extended Abstract)

Kim G. Larsen¹ and Axel Legay²
¹ Aalborg University, Denmark
kgl@cs.aau.dk
² INRIA/IRISA, France
axel.legay@inria.fr

Abstract. This extended abstract offers a brief survey presentation of the specification formalism of modal transition systems and its recent extensions to the quantitative setting of timed as well as stochastic systems. Some applications will also be briefly mentioned.

1 Modal Transition Systems: The Origins

Modal transition systems [46] provides a behavioural compositional specification formalism for reactive systems. They grew out of the notion of relativised bisimulation, which allows for simple specifications of components by allowing the notion of bisimulation to take the restricted use that a given context may have in its.

A modal transition system is essentially a (labelled) transition system, but with two types of transitions: so-called may transitions, that any implementation may (or may not) have, and must transitions, that any implementation must have. In fact, ordinary labelled transition systems (or implementations) are modal transition systems where the set of may- and must-transitions coincide. Modal transition systems come equipped with a bisimulation-like notion of (modal) refinement, reflecting that the more must-transitions and the fewer may-transitions a modal specification has the more refined and closer to a final implementation it is.

Example 1. Consider the modal transition system shown in Figure 1 which models the requirements of a simple email system in which emails are first received and then delivered – must and may transitions are represented by solid and dashed arrows, respectively. Before delivering the email, the system may check or process the email, e.g. for en- or decryption, filtering of spam emails, or generating automatic answers using as an auto-reply feature. Any implementation of this email system specification must be able to receive and deliver email, and it may also be able to check arriving email before delivering it. No other behavior is allowed. Such an implementation is given in Figure 2.

Modal transition systems play a major role in various areas. However, the model is best known by its application in compositional reasoning, which has been recognized in the ARTIST Network of Excellence and several other related European projects. In fact, modal transition systems have all the ingredients of a complete compositional

© IFIP International Federation for Information Processing 2013
specification theory allowing for logical compositions (e.g. conjunction) [42], structural compositions (e.g. parallel) [37] as well as quotienting permitting specifications of composite systems to be transformed into necessary and sufficient specification of components [35]. Thus, modal transition systems have all the benefits of both logical and behavioural specification formalisms [18]. Though modal refinement – like bisimulation – is polynomial-time decidable for finite-state modal transition systems, it only provides a sound but not complete test for the true semantic refinement between modal specification, in terms of set inclusion between their implementation-sets (so-called thorough refinement). For several years, the complexity of thorough refinement – as well as the consistency – between modal specifications was an open problem, which after a series of attempts [44, 3] [2] was shown to be EXPTIME-complete [14].

In the rest of this overview, we will briefly introduce several quantitative extensions of modal transition systems that have been recently proposed in the literature. Sections 2 to 5 mainly focus on modal transition systems as a specification theory, while Sections 6 and 7 outline some other extensions.

2 Timed Modal Specifications

It is well acknowledged that real-time can be a crucial parameter in practice, for example in embedded systems. This motivates the study of extended modal transition systems to introduce real-time features.

Timed extensions of modal transitions were introduced early on [23] as timed extension of the process algebra CCS. Unfortunately the supporting tool EPSILON was entirely relying on the so-called region-abstraction, making scalability extremely poor. Most recently, taking advantage of the powerful game-theoretical engine of UPPAAL
Tiga [10,21] a “modal-transition system”-like compositional specification theory based on Timed I/O Automata [41] has been proposed [24]. ECDAR [25] gives an efficient tool support for refinement, consistency and quotienting for this theory.

In [27], de Alfaro et al. suggested timed interfaces, a model that is similar to the one of TIOTSs. Our definition of composition builds on the one proposed there. However, the work in [27] is incomplete. Indeed, there was no notion of implementation and refinement. Moreover, conjunction and quotient were not studied. Finally, the theory has only been implemented in a prototype tool which does not handle continuous time, while our contribution takes advantage of the powerful game engine of UPPAAL Tiga.

The work of [16] suggests an alternative timed extension of modal transition systems (though still relying on regions for refinement algorithms). This work is less elaborated and implementation was not considered there.

3 Weighted Modal Specifications

The previous section was concerned with modal transition systems whose implementations are timed automata. There are various other extensions of automata, among which one finds weighted automata, that are classical automata whose transitions are equipped with integer weights. In [40], modal transition systems were extended with integer intervals in order to capture and abstract weighted automata. The works also proposes structural and logical composition as well as refinement for the extended model. Latter, in [7], the work was generalized to weighted modal transition systems, whose transition can be equipped with any type of quantity.

Albeit the extensions mentioned above allow for a quantitative treatment of automata behaviors, the operations on weighted modal transition systems remain qualitative. Especially, the refinement relation of modal transition systems is qualitative in nature, i.e. an implementation does, or does not, refine a specification. Such a view may be fragile in the sense that the inevitable approximation of systems by models, combined with the fundamental unpredictability of hardware platforms, make difficult to transfer conclusions about the behavior to the actual system. Hence, this approach is arguably unsuited for modern software systems. In [5], the first quantitative extension of modal automata was proposed. This model allows to capture quantitative aspects during the refinement and implementation process, thus leveraging the problems of the qualitative setting.

In [5], satisfaction and refinement are lifted from the well-known qualitative setting to the quantitative setting, by introducing a notion of distance between weighted modal transition systems. It is also shown that quantitative versions of parallel composition, as well as quotient (the dual operator to parallel composition), inherit the properties from the Boolean setting.

**Example 2 (taken from [5]).** Consider again the modal transition system of Figure 1, but this time with quantities, see Figure 3: Every transition label is extended by an integer intervals modeling upper and lower bounds on the time required for performing the corresponding actions. For instance, the reception of a new email (action receive) must take between one and three time units, the checking of the email (action check) is allowed to take up to five time units.
In this quantitative setting, there is a problem with using a Boolean notion of refinement: If one only can decide whether or not an implementation refines a specification, then the quantitative aspects get lost in the refinement process. As an example, consider the email system implementations in Figures 4 to 7. Implementation (a) does not refine the specification, as there is an error in the discrete structure of actions: after receiving an email, the system can check it indefinitely without ever delivering it. Also, implementations (b) and (c) do not refine the specification: (b) takes too long to receive any email, while (c) does not deliver the email fast. Implementation (d), on the other hand, is a perfect refinement of the specification.

The work in [5] uses an accumulating distance, but the contribution was latter generalized to any type of distance in [6]. The extended model allowed, as an example, to define various notions of robustness for specification theories. Complexity of refinement and efficient implementations remain open problems.

4 Probabilistic Modal Specifications

In [39], modal transitions systems were extended into a specification formalism for Markov Chains by the introduction of so-called probabilistic specifications (now known
as Interval Markov Chains), where concrete probabilities are replaced with intervals, and with refinement providing a conservative extension or probabilistic bisimulation [45]. However, Interval Markov Chains lack several of the properties required for a complete compositional specification theory; in particular, they are not closed neither under logical nor structural composition. Recently, the extended notion of Constraint Markov Chains [19] was introduced precisely with the purpose of providing these closure properties. A Constraint Markov Chain (CMC) is a Markov Chain (MC) equipped with a constraint on the next-state probabilities from any state. Roughly speaking, an implementation for a CMC is thus a MC, whose next-state probability distribution satisfies the constraint associated with each state. The power of constrains can be exploited to obtain closure under any logical/structural composition operation. The complexity of the refinement relation largely depends on the one to solve the constraints – it is at least quadratic (resp. exponential) for syntactic (resp. thorough) refinement. The reader interested in decision problems for CMCs is redirected to [32,31].

More recently, the concept of CMC was extended to offer abstractions for Probabilistic Automata (PA), i.e., structures that mix both stochastic and non-deterministic aspects. The work in [28] proposes Abstract Probabilistic Automata, that are a combination of modal transition systems and CMCs, modalities being used to capture the non-determinism in PAs. The model was implemented in the APAC toolset [30] and various decision problems, including stuttering and abstraction, were studied in [29,54].

Example 3 (taken from [29]). Consider the implementation (left) and specification (right) of a coffee machine given in Figure 8. The specification indicates that there are two possible transitions from initial state $I$: a may transition labeled with action $r$ (reset) and a must transition labeled with action $c$ (coin). May transitions are represented with dashed arrows and must transitions are represented with plain arrows. The probability distributions associated with these actions are specified by the constraints $\varphi_r$ and $\varphi_c$, respectively.

5 Beyond Modalities

In a seminal paper [26], de Alfaro and Henzinger promoted another specification theory known under the name of interface automata. An interface is represented by an input/output automaton [47], i.e., an automaton whose transitions are labeled with input or output actions. The semantics of such an automaton is given by a two-players
game: an *Input* player represents the environment, and an *Output* player represents the component itself. Interface automata do not encompass any notion of model, because there is no way to distinguish between an interface and its implementations.

Refinement between interface automata corresponds to the alternating refinement relation between games \[1\], i.e., an interface refines another if its environment is more permissive, whereas its component is more restrictive. Contrary to most interfaces theories, the game-based interpretation offers an *optimistic* treatment of composition: two interfaces can be composed if there exists at least one environment (i.e., one strategy for the *Input* player) in which they can interact together in a safe way (i.e., whatever the strategy of the *Output* player is). This is referred to as the compatibility of interfaces. A quotient, which is the adjoint of the game-based composition, has been proposed in \[17\] for the deterministic case.

In \[43,52\], modal transition systems and interface automata were combined to give rise to modal interfaces, a model that offers both the power of modalities and the optimistic composition approach of interface automata through labeling of may and must modalities with input/output features. Implementations of modal interfaces can be find in the Mika toolset \[20\] and the MIOs workbench \[9\].

It is worth mentioning that the methodology used in \[52\] is rather generic and can be applied to other extensions of modal automata. As an example, the APA model was also extended to a modal APA model in \[29\].

6 Contract Theory

We already observed that modal transition systems act as a good model for a complete specification theory. Of course, there are other similar approaches. In fact, some of them advocate that specification theories should be equipped with additional structure that makes more explicit their possible connections. This is particularly the case of the contract theory approach, which is based on a assume-guarantee (AG) reasoning.

Concretely, contract theories differ from classical specification theories in that they strictly follow the principle of separation of concerns. They separate the specification of assumptions from the specification of guarantees, a choice largely inspired by early ideas on manual proof methods by Misra, Chandy \[49\] and Jones \[38\], along with the wide acceptance of the pre-/post-condition specification in programming \[48,55\], and more in general semantical rules independent from language representation \[22\].

Recently, Benveniste et al \[15\] proposed a contract theory where assumptions and guarantees are represented by trace structures. While this work is of clear interest, it suffers from the absence of an effective representation for the embedded interface theory. Some extensions, such as the one proposed in \[50,35\], leverage this problem but in a specific manner, i.e., just for the case of a single theory.

In \[4\], it was shown how a theory of contracts can be built on top of a given abstract specification theory. Contracts are just pairs \((A, G)\) of an assumption and a guarantee specification. Particularly, it was shown how the contract theory can be instantiated by using modal transition systems.
7 Others Modal Extensions and Applications

There are many other extensions of modal transition systems, which include those that encompass unbounded data or costs and parameters [8, 12, 13], and those that offer a more elaborated treatment of modalities [5, 11].

In addition to their contribution to specification theories, modal transition systems have also played a major role in abstraction-based model checking [33, 34], software differences [53], and in the design of efficient approaches for software product lines verification [36].

References