Bibliography of Team Automata

Maurice H. ter Beek

February 10, 2017

References


Abstract: This paper introduces a mathematical model of groupware systems called Team Automata. We use this model at the architectural level, to describe components of a groupware system, and their interconnections. The multiple automata constituting the team interact via shared actions - transitions which multiple automata perform synchronously together. The paper concludes by illustrating the application of this model to the analysis of real time shared application groupware. The model suggests that there are design alternatives for shared application groupware which have mostly gone unexplored.


Abstract: Team automata have been proposed in [1] as a formal framework for modeling both the conceptual and the architectural level of groupware systems. Here we define team automata in a mathematically precise way in terms of component automata which synchronize on certain executions of actions.

At the conceptual level, our model serves as a formal framework in which such basic groupware notions as cooperation and collaboration can be rigorously defined and studied. At the architectural level, team automata can be used as building blocks in the design of groupware systems.
Abstract: Coordinating the efforts of multiple teams working in parallel on a model is a non-trivial task. A major part of this effort is to resolve conflicts, which are only detected when the work of the separate teams is integrated. In this paper we discuss how a model can be cut into distinct packages where in parallel each of these packages is locally modified by just one of the teams. Integration of the modified packages is straightforward as we only allow local changes to a package, i.e. changes that do not propagate beyond the package and that do not cause conflicts during integration. Additionally, we show how the package structure of a model and the teams working on the packages can be (temporarily) adapted to manage the need for non-local changes. We model the teams by team automata and discuss how their possible errant behaviour, which can lead to conflicts, is restricted by our strategy of model development.

Abstract: Team automata are a mathematical modeling tool for capturing notions of coordination, collaboration, and cooperation. They are based upon the concept of “shared actions”. In this document we explain the team automata model in the context of access control mechanisms. This demonstrates the model usage and utility for capturing information security and protection structures, and critical coordinations between these structures. Modeling spatial access mechanisms as team automata not only allows analysis and verification of certain properties, it also suggests some extensions to the usual notions of access control.

Abstract: Team automata have been proposed as a formal framework for modelling both the conceptual and the architec-
Team automata have been proposed as a formal framework for modelling both the conceptual and the architectural level of groupware systems. They are defined in terms of component automata together with an interconnection mechanism which is based on shared actions (synchronizations). Components can be combined in a loose or more tight fashion depending on which actions are to be shared, and when. The formal setup makes it possible to distinguish between, e.g., master-slave and peer-to-peer synchronizations and to classify team automata based on the mode of synchronization. Since a team automaton can be used as a component in a higher-level team, the framework allows for the representation of hierarchical systems. As an example, using a spatial access metaphor, we will consider some access control strategies in the context of team automata.


Abstract: Team automata have been proposed as a formal framework for modelling both the conceptual and the architectural level of groupware systems. They are defined in terms of component automata together with an interconnection mechanism which is based on shared actions (synchronizations). Components can be combined in a loose or more tight fashion depending on which actions are to be shared, and when. The formal setup makes it possible to distinguish between, e.g., master-slave and peer-to-peer synchronizations and to classify team automata based on the mode of synchronization. Since a team automaton can be used as a component in a higher-level team, the framework allows for the representation of hierarchical systems. As an example, using a spatial access metaphor, we will consider some access control strategies in the context of team automata.


Abstract: Team automata provide a framework for capturing notions like coordination, collaboration, and cooperation in distributed systems. They consist of an abstract specification of components of a system and allow one to describe different interconnection mechanisms based upon the concept of “shared
This document considers access control mechanisms in the context of the team automata model. It demonstrates the model usage and utility for capturing information security and protection structures, and critical coordinations between these structures. On the basis of a spatial access metaphor, various known access control strategies are given a rigorous formal description in terms of synchronizations in team automata.


Abstract: The paper studies and compares two different approaches to model communication and cooperation. The approaches are team automata, a well-defined variant of communicating automata, and state-charts, heavily used in object-oriented modelling methods. The comparison yields interesting insights for modelling communication and cooperation. In particular, the differences between action-based, synchronous and state-based, asynchronous communication are elucidated.


Abstract: We introduce team pushdown automata as a theoretical framework capable of modelling various communication and cooperation strategies in multi-agent systems. Team pushdown automata are obtained by augmenting distributed pushdown automata with the notion of team cooperation or - alternatively - by augmenting team automata with pushdown memory. In a team pushdown automaton, several pushdown automata work as a team on the input word placed on a common one-way input tape. At any moment in time one team of pushdown automata, each with the same symbol on top of its stack, is active: each pushdown automaton in the active team replaces the topmost symbol of its stack and changes state, while the current input symbol is read from the input tape by a common reading head. The teams are formed according to the team cooperation strategy of the team pushdown automaton and may vary from one moment to the other. Based on the notion of competence, we introduce a variety of team cooperation strategies. If all stacks
are empty when the input word has been completely read, then this word is part of the language accepted by the team pushdown automaton. Our initial focus is on the accepting capacity of team pushdown automata. We conclude by providing some directions for future work, including a hint at an application of the enhanced modelling power of team automata obtained through the addition of pushdown memory.


Abstract: Team automata have been proposed in Ellis (1997) as a formal framework for modeling both the conceptual and the architectural level of groupware systems. Here we define team automata in a mathematically precise way in terms of component automata which synchronize on certain executions of actions. At the conceptual level, our model serves as a formal framework in which basic groupware notions can be rigorously defined and studied. At the architectural level, team automata can be used as building blocks in the design of groupware systems.


Abstract: We show that team automata (TA) are well suited to model secure multicast/broadcast communication with possible packet loss. This is a consequence of the natural way in which one-to-many (one-to-all) transmissions typical of multicast (broadcast) sessions can be modelled as communications between the component automata (CA) constituting a TA. To this aim, we use TA to model an instance of the EMSS multicast protocol family. In addition, we investigate a formulation of the Generalized Non-Deducibility on Compositions (GNDC) schema in terms of TA with the aim to embed TA in this well-established analysis framework. We intend to use this new setting for the formal verification of security properties for stream signature protocols.

Abstract: We show that team automata (TA) are well suited to model secure multicast/broadcast communication with possible packet loss. This is a consequence of the natural way in which one-to-many (one-to-all) transmissions typical of multicast (broadcast) sessions can be modelled as communications between the component automata (CA) constituting a TA. To this aim, we use TA to model an instance of the EMSS multicast protocol family. In addition, we investigate a formulation of the Generalized Non-Deducibility on Compositions (GNDC) schema in terms of TA with the aim to embed TA in this well-established analysis framework. We intend to use this new setting for the formal verification of security properties for stream signature protocols.


Abstract: A team automaton is said to satisfy compositionality if its behaviour can be described in terms of the behaviour of its constituting component automata. As an initial investigation of the conditions under which team automata satisfy compositionality, we study their computations and behaviour in relation to those of their constituting component automata. We show that the construction of team automata according to certain natural types of synchronization guarantees compositionality.


Abstract: A team automaton is said to satisfy compositionality if its behaviour can be described in terms of the behaviour of its constituting component automata. As an initial investigation of the conditions under which team automata satisfy compositionality, we study their computations and behaviour in relation to those of their constituting component automata. We show that the construction of team automata according to certain natural types of synchronization guarantees compositionality.
Abstract: This thesis studies formal aspects of team automata, a mathematical framework introduced in 1997 by Ellis to model components of groupware systems and their interconnections. We focus on the flexibility team automata offer when modeling collaboration between system components.

Abstract: Team automata have been proposed as a formal framework for modelling both the conceptual and the architectural level of groupware systems. They are defined in terms of component automata (labelled transition systems) together with an interconnection mechanism which is based on shared actions (synchronizations). Components can be combined in different fashions depending on which actions are to be shared, and when. This set-up makes it possible to distinguish between different types of synchronizations and allows to represent hierarchical systems. A survey is presented including a brief comparison with some related models.

Abstract: We introduce team pushdown automata as a theoretical framework capable of modelling various communication and cooperation strategies in complex, distributed systems. Team pushdown automata are obtained by augmenting distributed pushdown automata with the notion of team cooperation or - alternatively - by augmenting team automata with pushdown memory. Here we study their accepting capacity.

Abstract: We introduce team pushdown automata as a theoretical framework capable of modelling various communication and cooperation strategies in complex, distributed systems. Team pushdown automata are obtained by augmenting distributed pushdown automata with the notion of team cooperation or - alternatively - by augmenting team automata with pushdown memory. Here we study their accepting capacity.
Abstract: We introduce team pushdown automata (PDAs) as a theoretical framework capable of modelling various communication and cooperation strategies in complex, distributed systems. Team PDAs are obtained by augmenting distributed PDAs with the notion of team cooperation or, alternatively, by augmenting team automata with pushdown memory. In a team PDA, several PDAs work as a team on the input word placed on a common one-way input tape. At any moment in time one team of PDAs, each with the same symbol on top of its stack, is active: each PDA in the active team replaces the topmost symbol of its stack and changes state, while the current input symbol is read from the input tape by a common reading head. The teams are formed according to the team cooperation strategy of the team PDA and may vary from one moment to the other. Based on the notion of competence, we introduce a variety of team cooperation strategies. If all stacks are empty when the input word has been completely read, then this word is part of the language accepted by the team PDA. Here we focus on the accepting capacity of team PDA.


Abstract: We show that team automata (TA) are well suited for security analysis by reformulating the Generalized Non-Deducibility on Compositions (GNDC) schema in terms of TA. We then use this to show that integrity is guaranteed for a case study in which TA model an instance of the Efficient Multi-chained Stream Signature (EMSS) protocol.


Abstract: Considered are reactive systems consisting of components which cooperate through synchronizations on common actions. Within the framework of team automata such systems may be constructed by iteratively adding components. Our aim is to formulate conditions on the components which would guarantee correct (interactive) behaviour of such hierarchically constructed systems. As a possible solution an extension of the (binary) notion of input/output compatibility is proposed.

**Abstract:** Considered are reactive systems consisting of components which cooperate through synchronizations on common actions. Within the framework of team automata such systems may be constructed by iteratively adding components. Our aim is to formulate conditions on the components which would guarantee correct (interactive) behaviour of such hierarchically constructed systems. As a possible solution an extension of the (binary) notion of input/output compatibility is proposed.


**Abstract:** In [17], Kleijn presented a survey of the use of team automata for the specification and analysis of phenomena from the field of computer supported cooperative work, in particular notions related to groupware systems. In this paper we present a survey of the use of team automata for the specification and analysis of some issues from the field of security.


**Abstract:** We use Team Automata in order to model a protocol for securing agents in a hostile environment. Our study focuses on privacy properties of the agents. We use the framework to prove a result from Kilian et al. As a by-product, our analysis gives some initial insight on the limits of the protocol. From a different perspective, this study continues a line of research on the expressive power and modelling capabilities of Team Automata. To the best of our knowledge, this is the first attempt to use Team Automata for the analysis of privacy properties.


**Abstract:** In [17], Kleijn presented a survey of the use of team automata for the specification and analysis of phenomena from
the field of computer supported cooperative work, in particular notions related to groupware systems. In this paper we present a survey of the use of team automata for the specification and analysis of some issues from the field of security. In particular, we show how team automata can adequately be used to model and verify various access control policies, multicast/broadcast communication protocols, and general (cryptographic) communication protocols.


Abstract: It is shown how Input/Output automata fit in the framework of team automata, thus making it possible to view certain notions and results regarding their modular structure as special instances of more general observations.


Abstract: (Chapter 4: The Team Automata Chapter) This chapter is devoted to modeling and analysis in the framework of team automata. A first part describes a relevant multicast security protocol and a secure protocol for mobile agents, by means of team automata. New analysis strategies within team automata are then presented. Finally, we analyze the modelled protocols, by means of the above-cited strategies. The Team Automata chapter contributes towards the analysis of procedures with non finite behaviour with respect to the number of participants. Furthermore, it gives the first attempt to analyze security and privacy properties by team automata. An automaton framework is adopted.


Abstract: In [17], Kleijn presented a survey of the use of team automata for the specification and analysis of phenomena from the field of computer supported cooperative work, in particular notions related to groupware systems. In this paper we present a survey of the use of team automata for the specification and analysis of some issues from the field of security. In particular, we show how team automata can adequately be used to model
and verify various access control policies, multicast/broadcast communication protocols, and general (cryptographic) communication protocols.


**Abstract:** None.


**Abstract:** (Chapter 6: Security Analysis with Team Automata)
In this chapter we develop a framework based on team automata that can be used for formal security analysis. To this aim, we first define an insecure communication scenario for team automata, which is general enough to encompass various communication protocols. Then, we reformulate the Generalized Non-Deducibility on compositions schema, originally introduced in the context of process algebras, in terms of team automata. Based on the resulting framework, we subsequently develop a compositional analysis strategy that can be used for the verification of security properties for a variety of communication protocols. We apply the framework in practise, by showing that integrity is guaranteed for a particular instance of the Efficient Multi-chained Stream signature protocol.


**Abstract:** Team automata are a formalism for the component-based specification of reactive, distributed systems. Their main feature is a flexible technique for specifying coordination patterns among distributed systems, extending classical I/O automata. Furthermore, for some of these patterns, the language recognised by a team automaton can be specified in terms of the languages recognised by its components. The present paper introduces a process calculus tailored over team automata. Each automaton is thus described by a process, in such a way that its associated (fragment of a) labelled transition system is bisimilar to the original automaton. Furthermore, the mapping is proved to be denotational, since the operators on processes are in a bijective correspondence with a chosen family of coordination patterns,
and that correspondence is preserved by the mapping. The paper thus extends to team automata some classical results on I/O automata and their representation by process calculi. Moreover, besides providing a language for expressing team automata and their composition, we widen the family of coordination patterns for which an equational characterisation of the language associated to a composite automaton can be provided. The latter result is obtained by providing a set of axioms, in ACP-style, for capturing bisimilarity in our calculus.


Abstract: We use Team Automata in order to model a protocol for securing agents in a hostile environment. Our study focuses on privacy properties of the agents. We use the framework to prove a result from Kilian et al. As a by-product, our analysis gives some initial insight on the limits of the protocol. From a different perspective, this study continues a line of research on the expressive power and modelling capabilities of Team Automata. To the best of our knowledge, this is the first attempt to use Team Automata for the analysis of privacy properties.


Abstract: Team automata are a formalism for the component-based specification of reactive, distributed systems. Their main feature is a flexible technique for specifying coordination patterns among systems, thus extending I/O automata. Furthermore, for some patterns the language recognized by a team automaton can be specified via those languages recognized by its components. We introduce a process calculus tailored over team automata. Each automaton is described by a process, and such that its associated (fragment of a) labeled transition system is bisimilar to the original automaton. The mapping is furthermore denotational, since the operators defined on processes are in a bijective correspondence with a chosen family of coordination patterns and that
correspondence is preserved by the mapping. We thus extend to
team automata a few classical results on I/O automata and their
representation by process calculi. Moreover, besides providing a
language for expressing team automata, we widen the family of
coordination patterns for which an equational characterization of
the language associated to a composite automaton can be pro-
vided. The latter result is obtained by providing a set of axioms,
in ACP-style, for capturing bisimilarity in our calculus.

[34] M.H. ter Beek, G. Lenzini, and M. Petrocchi. A Team Automaton Scen-
ario for the Analysis of Security Properties in Communication Protocols.
URL: http://fmt.isti.cnr.it/mtbeek/jalc.pdf

Abstract: Formal methods are a popular means to specify and
verify security properties of a variety of communication protocols.
In this article we take a step towards the use of team automata
for the analysis of security aspects in such protocols. To this aim,
we define an insecure communication scenario for team automata
that is general enough to encompass various communication pro-
tocols. We then reformulate the Generalized Non-Deducibility
on Compositions schema—originally introduced in the context of
process algebras—in terms of team automata. Based on the re-
sulting team automata framework, we subsequently develop two
analysis strategies that can be used to verify security properties
of communication protocols. Indeed, the paper concludes with
two case studies in which we show how our framework can be
used to prove integrity and secrecy in two different settings: We
show how integrity is guaranteed in a team automaton model
of a particular instance of the Efficient Multi-chained Stream
Signature protocol, a communication protocol for signing digital
streams that provides some robustness against packet loss, and
we show how secrecy is preserved when a member of a multicast
group leaves the group in a particular run of the complementary
variable approach to the $N$-Root/Leaf pairwise keys protocol.

ware Architectures Using Extended Automata-Based Models. In F. Arbab
and M. Sirjani, editors, Proceedings of the International Symposium on
Fundamentals of Software Engineering (FSEN’07). Tehran, Iran, volume
4767 of Lecture Notes in Computer Science, pages 423–431. Springer-
Verlag, Berlin, 2007. URL: http://dx.doi.org/10.1007/978-3-540-75698-9_30

Abstract: Applying an appropriate formal model to specify soft-
ware architecture makes a reliable foundation to formally ver-
ify non-functional properties and therefore, leads to early detec-
tion of defects. In this paper we make a comparison between
automata-based models and evaluate their abilities to model different aspects of components interaction in software architectures. We try to use Team automata as a middleware to formally specify well-known architectural descriptions in UML2.0. A Limitation of current automata models, so called “actions interleaving” is also discussed and some approaches to overcome this limitation described.


Abstract: Team automata are a formalism for the component-based specification of reactive, distributed systems. Their main feature is a flexible technique for specifying coordination patterns among systems, thus extending I/O automata. Furthermore, for some patterns the language recognized by a team automaton can be specified via those languages recognized by its components. We introduce a process calculus tailored over team automata. Each automaton is described by a process, such that its associated (fragment of a) labeled transition system is bisimilar to the original automaton. The mapping is moreover denotational, since the operators defined on processes are in a bijective correspondence with a chosen family of coordination patterns and that correspondence is preserved by the mapping. We thus extend to team automata a few classical results on I/O automata and their representation by process calculi. Moreover, besides providing a language for expressing team automata, we widen the family of coordination patterns for which an equational characterization of the language associated to a composite automaton can be provided. The latter result is obtained by providing a set of axioms, in ACP-style, for capturing bisimilarity in our calculus.


Abstract: Team automata, as an extended automata-based model, possesses distinguishable characteristics which make it suitable to formally specify and evaluate software architectural design. In this paper, we describe the benefits of TA over similar
automata models and show how can extend it to specify and evaluate performance of components interaction in Software Architectures. An application system example of using the introduced approach is also presented.


**Abstract:** Motivated by different ways to obtain team automata from synchronizing component automata, we consider various definitions of synchronized shuffles of words. A shuffle of two words is an interleaving of their symbol occurrences which preserves the original order of these occurrences within each of the two words. In a synchronized shuffle, however, also two occurrences of one symbol, each from a different word, may be identified as a single occurrence. In case at least one of the words involved is infinite, a (synchronized) shuffle can also be unfair in the sense that an infinite word may prevail from some point onwards even when the other word still has occurrences to contribute to the shuffle. We prove that for the synchronized shuffle operations under consideration, every (fair or unfair) synchronized shuffle can be obtained as a limit of synchronized shuffles of the finite prefixes of the words involved. In addition, it is shown that with the exception of one, all synchronized shuffle operations that we consider satisfy a natural notion of associativity, also in case of unfairness. Finally, using these results, some compositionality results for team automata are established.


**Abstract:** In this paper a framework called Team Automata (TA) has been introduced for capturing notions like coordination, collaboration and cooperation in distributed systems. It consists of component automata, combined in a coordinated way such that they can perform shared actions. Moreover, we consider a Spatio-Temporal Role Based Access Control Model (STRBAC) in the context of the team automata model, it describes the usage of the model and utility for capturing information security structures as well as critical coordination between these structures on the basis of Spatio-Temporal metaphor.
this work, Known access control strategies are given a formal
description in terms of synchronization in TA.


**Abstract:** Vector team automata are team automata with an explicit representation of synchronizations. This makes a translation possible of a subclass of vector team automata into individual token net controllers, a model of labeled Petri nets developed within the framework of vector controlled concurrent systems.


**Abstract:** A distributed environment where many components interact may be functioning in a suboptimal manner due to two main factors: message loss and deadlocks. Message loss occurs when a component is not ready to receive as input a message sent to it. In the case of a deadlock, a system is indefinitely waiting for a message that never arrives. In Carmona and Cortadella (2002) a theory has been presented for characterizing when a pair of systems is compatible in the sense that they can engage in a dialog free from these two problems. The theory developed was restricted to only two components, a particular mode of synchronization and a closed environment. In this paper we lift all these assumptions to define a general notion of compatibility in a multi-component environment. For the extended definition of compatibility, we use team automata as a modeling formalism which allows arbitrary synchronization strategies and iterative/hierarchical composition. Moreover, it is shown how the general definition of compatibility presented in this paper can be used to determine the compatibility problems that arise in a team automaton built on the basis of an arbitrary synchronization strategy.


**Abstract:** In CD grammar systems, the rewriting process is distributed over component grammars that take turns in the derivation of new symbols. Team automata however collaborate by synchronising their actions. Here we investigate how to transfer this concept of synchronisation to grammars by defining grammar teams that agree on the generation of shared terminal symbols based on a novel notion of competence. We first illustrate this
idea for the case of regular grammars and next propose an extension to the case of context-free grammars.

http://dx.doi.org/10.1007/978-3-319-47166-2_55

Abstract: We consider systems composed of reactive components that collaborate through synchronised execution of common actions. These multi-component systems are formally represented as team automata, a model that allows a wide spectrum of synchronisation policies to combine components into higher-level systems. We investigate the correct-by-construction engineering of such systems of systems from the point of view of correct communications between the components (no message loss or deadlocks due to indefinite waiting). This leads to a proposal for a generic definition of compatibility of components relative to the adopted synchronisation policy. This definition appears to be particularly appropriate for so-called master-slave synchronisations by which input actions (for ‘slaves’) are driven by output actions (from ‘masters’).