Input/Output Types for Dynamic Web Data (Extended Abstract)

Svetlana Jakšić

Faculty of Technical Sciences
University of Novi Sad
sjaksic@uns.ac.rs

As information networks become more open and dynamic, the need for protecting security and privacy of data is increasingly important in many fields of human activities. Systems must be able to exchange data and processes while preserving security. In case we are given a target security policy for a distributed system containing XML data, how can we check weather the system behaves according to the policy? One solution is to suitably annotate the security relevant events, to classify them according to a type system and to verify security properties by typing. In [3] we introduced ®X$d\pi$-calculus, a calculus for role-based access control of dynamic web data, and a type system for it and we verify the security properties by typing. Here, a subtyping relation which extends the type system for ®X$d\pi$ is proposed.

First, an example which illustrates the proposed approach and advantages of the extended type system will be given and then a brief presentation of the subtyping relation and a discussion of the related work. The full presentation of ®X$d\pi$-calculus and the corresponding type system, which can be found [3], will be omitted here.

An example

Let us consider a simple distributed system consisting of code running on behalf of four principals: an online discussion forum, a guest, a member and the moderator. Let the forum, written in XML notation, have the shape:

$$
\begin{array}{l}
< \texttt{forum} > \\
\quad < \texttt{general} > \\
\quad\quad forum\ rules \\
\quad < /\texttt{general} > \\
\quad < \texttt{music} > \\
\quad\quad lyrics \\
\quad < /\texttt{music} > \\
< /\texttt{forum} >
\end{array}
$$

The forum contains two main topics: the `general` and the `music`. In order to describe the behaviour of the guest, the member and the moderator we will use process calculus notation. In the syntax of ®X$d\pi$-calculus we consider four kinds of processes: π-calculus processes [7], for modelling local communication; `go` command, for modelling process migration between locations, as in $D\pi$ calculus [6]; `run`, `read` and `change` commands and for modelling interaction of processes with local data in place of the

update command of [5] and commands enable and disable for changing permissions to access data. We write $\texttt{read}_{\texttt{forum/general}}(\chi)$ for a guest wishing to read forum rules, where χ is a data pattern he is looking for, and $\texttt{read}_{\texttt{forum/music}}(\chi)$ for a member wishing to read the lyrics. The process $!b(y).\texttt{change}_{\texttt{forum}}(x, x|y)$ represents the moderator of the forum who has the ability to add a new topic. The moderator receives the new topic, updates the forum with it and waits to receive the next topic. In [3] we have investigated a system in which different participants can have different rights. We have achieved diversity and control of the rights by introducing role-based access control. More precisely, each tag is assigned a set of roles that a process is required to have in order to access it. The forum of this example decorated with sets of roles is:

$$
\begin{array}{l}
< \texttt{forum role} = \texttt{guest} > \\
\quad < \texttt{general role} = \texttt{guest} > \\
\quad\quad forum\ rules \\
\quad < /\texttt{general} > \\
\quad < \texttt{music role} = \texttt{member} > \\
\quad\quad lyrics \\
\quad < /\texttt{music} > \\
< /\texttt{forum} > .
\end{array}
$$

The same forum written in the syntax of our calculus is:

$$\texttt{forum}^{\{\texttt{guest}\}}[\texttt{general}^{\{\texttt{guest}\}}[forum\ rules]|[\texttt{music}^{\{\texttt{member}\}}[lyrics]].$$

Let the roles guest, member and moderator belong to a countable set of roles which is a lattice for a partial order $\sqsubseteq$. We consider guest $\sqsubseteq$ member $\sqsubseteq$ moderator. As expected, we assign the role guest to the unregistered guest of the forum, the role member to the registered user and the role moderator to the moderator of the forum. We say that the tag (or the edge when we use tree representation of XML documents) forum is accessible to the process with role guest or higher. The path forum/general is accessible to the process with the role guest since both tags are accessible to it, while the path forum/music is not. The forum we have described here is a "wiki" forum that allows guests and members to add content to the parts they have access to, as on an Internet forum, but also allows them to edit the content. All processes belonging to the same role have the same rights. Locations contain the processes and data. The behaviour of all the principals in the system is controlled with location policies and type system introduced in [3]. The policy of a location regulates changes of access rights. For example, if the forums' location policy is $(\{\texttt{guest}\}, \{(\{\texttt{moderator}\}, \texttt{guest})\}, \{(\{\texttt{moderator}\}, \texttt{member})\})$ then the processes with a role lower then guest can not access the forum at all and that the moderator may allow guest to access more topics or ban members to access the some topics in the forum. The dynamic change of access rights to data is done by adding or removing roles from the sets of roles on the data tree edges. The type system checks if a data tree and a process conform to a given location policy.

We propose an extension of the type system of [3] with subtyping relation in order to describe richer behaviour in our model. In the forum example, the process

$$
\begin{array}{c}
\bar{b}\langle \texttt{new}^{\{\texttt{guest}\}}[\ldots]\rangle^{\neg\{\texttt{guest}\}} \mid \bar{b}\langle \texttt{new}^{\{\texttt{member}\}}[\ldots]\rangle^{\neg\{\texttt{member}\}} \\
\mid\ !b(y).\texttt{change}_{\texttt{forum}^{\{\texttt{moderator}\}}}(x, x|y)^{\neg\{\texttt{moderator}\}}
\end{array}
$$

which represents a guest and a member, both wishing to send a new topic to the moderator for approval, is rejected by the type system of [3]. However, with the proposed subtyping relation, this process is typable.

Subtyping relation

We assume a countable set of roles $\mathcal{R}$, and use r to range over elements of $\mathcal{R}$. Let $(\mathcal{R}, \sqsubseteq)$ be a lattice and let $\bot, \top \in \mathcal{R}$ be its bottom and top element, respectively. By α, ρ, σ we denote non-empty sets of roles and by τ, ζ sets of roles containing the $\top$ element. We introduce a pre-order relation on value types in order to expand a domain of values that channels can communicate. With this aim, we have enriched the set of value types from [3] with the type of channels emitting values and with the type of channels receiving values as in [8, 9]. The types are presented in Table 1. Tv ranges over *value types* where as a value we consider either a channel name, a script, a location name, a path or a tree.

Table 1. The Syntax of ®X$d\pi$ Types

$Ch(Tv)$	type of channels communicating values of type Tv
$Ch^{!}(Tv)$	type of channels emitting values of type Tv
$Ch^{?}(Tv)$	type of channels receiving values of type Tv
$Loc(\mathcal{P})$	type of locations with the policy $\mathcal{P}$
$Script(\mathcal{P})$	type of scripts which can be activated at locations with the policy $\mathcal{P}$
$Path(\alpha)$	type of paths having the last edge with the set of roles α
$Pointer(\alpha)$	type of pointers whose path is typed by $Path(\alpha)$
$Tree(\mathcal{P}, \tau, \zeta)$	type of trees, which can stay at locations with the policy $\mathcal{P}$, with initial branches asking τ and which can be completely accessed by processes with at least one role of ζ
$Proc(\mathcal{P}, \rho)$	type of pure processes, which can stay at locations with the policy $\mathcal{P}$ and which can be assigned roles ρ
$ProcRole(\mathcal{P})$	type of processes with roles which can stay at locations with the policy $\mathcal{P}$

The subtyping rule

$$\frac{\Gamma \vdash v : Tv_1 \quad Tv_1 \prec Tv_2}{\Gamma \vdash v : Tv_2}$$

states that if Tv_1 is subtype of Tv_2, then a value of type Tv_1 is also of type Tv_2. The subtyping relation is such that if a *channel* can communicate values of type Tv then it can do both, emit and receive the values. Any channel that is receiving values of some type can be regarded as a channel receiving higher values. Any channel that is emitting values of some type can be regarded as a channel emitting lower values. The type of a *location* with the policy $\mathcal{P}_1$ is lower then the type of a location with policy $\mathcal{P}_2$ if $\mathcal{P}_2$ is less restrictive then $\mathcal{P}_1$. If a *script* can be activated at a location then it can also be

activated at any bigger location. A *path* having the last edge with a set of roles α_1 can be regraded as having at last edge any set of bigger or equal roles to those from α_1. Suppose that we are given a *tree* that can stay at a location of some policy $\mathcal{P}$, with initial branches asking τ_1 and that can be completely accessed by a process with at least one role from ζ_1. We can say that its initial branches are also asking set of roles that are greater than or equal to ζ_1 and it can be completely accessed by processes with roles greater or equal to ζ_1.

By extending the proof from [3], we can prove that the system satisfies the subject reduction and other relevant properties of well behaved processes. We proved in [3] that processes can communicate only values with at least one characteristic role lower than equal to a role of the process. The subtyping relation implies that channels emitting a value of type can also emit all the values that are of smaller type with respect to the relation. The channels receiving values of a type can also receive values that have bigger types.

Conclusions and related work

In this paper a notion of subtyping is added to the type system of the ®X$d\pi$-calculus and it is demonstrated that subtyping increases the flexibility of types. The type systems and calculi discussed here strongly relies on [3] and is most related to [4, 1] and [2]. Input and output types are those from [8, 9]

Acknowledgment. This work has been supported by the Serbian Ministry of Education and Science (projects ON174026 and III44006) and Provincial Secretariat for Science and Technological Development of Province of Vojvodina. ICTCS reviewers have provided useful comments.

References

1. Chiara Braghin, Daniele Gorla, and Vladimiro Sassone. Role-based access control for a distributed calculus. *Journal of Computer Security*, 14(2):113–155, 2006.
2. Adriana B. Compagnoni, Elsa L. Gunter, and Philippe Bidinger. Role-based access control for boxed ambients. *Theoretical Computer Science*, 398(1-3):203–216, 2008.
3. Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Svetlana Jakšic, and Jovanka Pantovic. Types for Role-Based Access Control of Dynamic Web Data. In *WFLP'10*, volume 6559 of *LNCS*, pages 1–29. Springer, 2011.
4. Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Jovanka Pantovic, and Daniele Varacca. Security types for dynamic web data. *Theoretical Computer Science*, 402(2-3):156–171, 2008.
5. Philippa Gardner and Sergio Maffeis. Modelling dynamic web data. *Theoretical Computer Science*, 342(1):104–131, 2005.
6. Matthew Hennessy and James Riely. Resource access control in systems of mobile agents. *Information and Computation*, 173(1):82–120, 2002.
7. Robin Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.
8. Benjamin C. Pierce and Davide Sangiorgi. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, 6(5):409–453, 1996.
9. Davide Sangiorgi and David Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.