Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 138 (2005) 23–42
www.elsevier.com/locate/entcs

Security Policies as Membranes in Systems for Global Computing[1]

Daniele Gorla[2]

Dip. di Informatica, Univ. di Roma "La Sapienza", Italy
Dip. di Sistemi ed Informatica, Univ. di Firenze, Italy

Matthew Hennessy[3]

Dep. of Informatics, Univ. of Sussex, Brighton (UK)

Vladimiro Sassone[4]

Dep. of Informatics, Univ. of Sussex, Brighton (UK)

Abstract

We propose a simple global computing framework, whose main concern is code migration. Systems are structured in sites, and each site is divided into two parts: a computing body, and a *membrane* which regulates the interactions between the computing body and the external environment. More precisely, membranes are filters which control access to the associated site, and they also rely on the well-established notion of *trust* between sites. We develop a basic theory to express and enforce security policies via membranes. Initially, these only control the actions incoming agents intend to perform locally. We then adapt the basic theory to encompass more sophisticated policies, where the number of actions an agent wants to perform, and also their order, are considered.

Keywords: Global Computing, Code Migration, Access Control, Security Policies, Types.

[1] This work has been partially supported by EU FET – Global Computing initiative, projects MIKADO IST-2001-32222 and MyThS IST-2001-32617. The funding bodies are not responsible for any use that might be made of the results presented here.
[2] Email: gorla@di.uniroma1.it
[3] Email: matthewh@susx.ac.uk
[4] Email: vs@susx.ac.uk

doi:10.1016/j.entcs.2005.05.003

1 Introduction

Computing is increasingly characterised by the global scale of applications and the ubiquity of interactions between mobile components. Among the main features of the forthcoming "global ubiquitous computing" paradigm we list *distribution* and *location awarness*, whereby code located at specific sites acts appropriately to local parameters and circumstances, that is, it is "context-aware"; *mobility*, whereby code is dispatched from site to site to increase flexibility and expressivity; *openness*, reflecting the nature of global networks and embodying the permeating hypothesis of localised, partial knowledge of the execution environment. Such systems present enormous difficulties, both technical and conceptual, and are currently more at the stage of exciting future prospectives than that of established of engineering practice. Two concerns, however, appear to clearly have a ever-reaching import: *security* and *mobility control*, arising respectively from openness and from massive code and resource migrations. They are the focus of the present paper.

We aim at classifying mobile components according to their behaviour, and at empowering sites with control capabilities which allow them to deny access to those agents whose behaviour does not conform to the site's *policy*. We see every site of a system

$$k[\![\, M \,\rangle\!\rangle\, P \,]\!]$$

as an entity named k and structured in two layers: a *computing body* P, where programs run their code – possibly accessing local resources offered by the site – and a *membrane* M, which regulates the interactions between the computing body and the external environment. An agent P wishing to enter a site N must be verified by the membrane before it is given a chance to execute in N. If the preliminary check succeeds, the agent is allowed to execute, otherwise it is rejected. In other words, a membrane implements the policy each site wants to enforce locally, by ruling on the requests of access of the incoming agents. This can be easily expressed by a migration rule of the form:

$$k[\![\, M^k \,\rangle\!\rangle\, \mathbf{go}\, l.P \mid Q \,]\!] \parallel l[\![\, M^l \,\rangle\!\rangle\, R \,]\!] \rightarrow k[\![\, M^k \,\rangle\!\rangle\, Q \,]\!] \;\parallel\; l[\![\, M^l \,\rangle\!\rangle\, P \mid R \,]\!] \qquad \text{if } M^l \vdash^k P$$

The relevant parts here are P, the agent wishing to migrate from k to l, and l, the receiving site, which needs to be satisfied that P's behaviour complies with its policy. The latter is expressed by l's membrane, M^l. The judgement $M^l \vdash^k P$ represents l inspecting the incoming code to verify that it upholds M^l.