Co-design Environment for Secure Embedded Systems

Matt Eby, Janos L. Mathe, Jan Werner, Gabor Karsai, Sandeep Neema, Janos Sztipanovits, Yuan Xue

Institute for Software Integrated Systems, Vanderbilt University

Incorporating Security into DSMLs

Advantages of Design Time Security Modeling

Domain Specific Modeling Language (DSML) examples

- SysML
- AADL
- UML

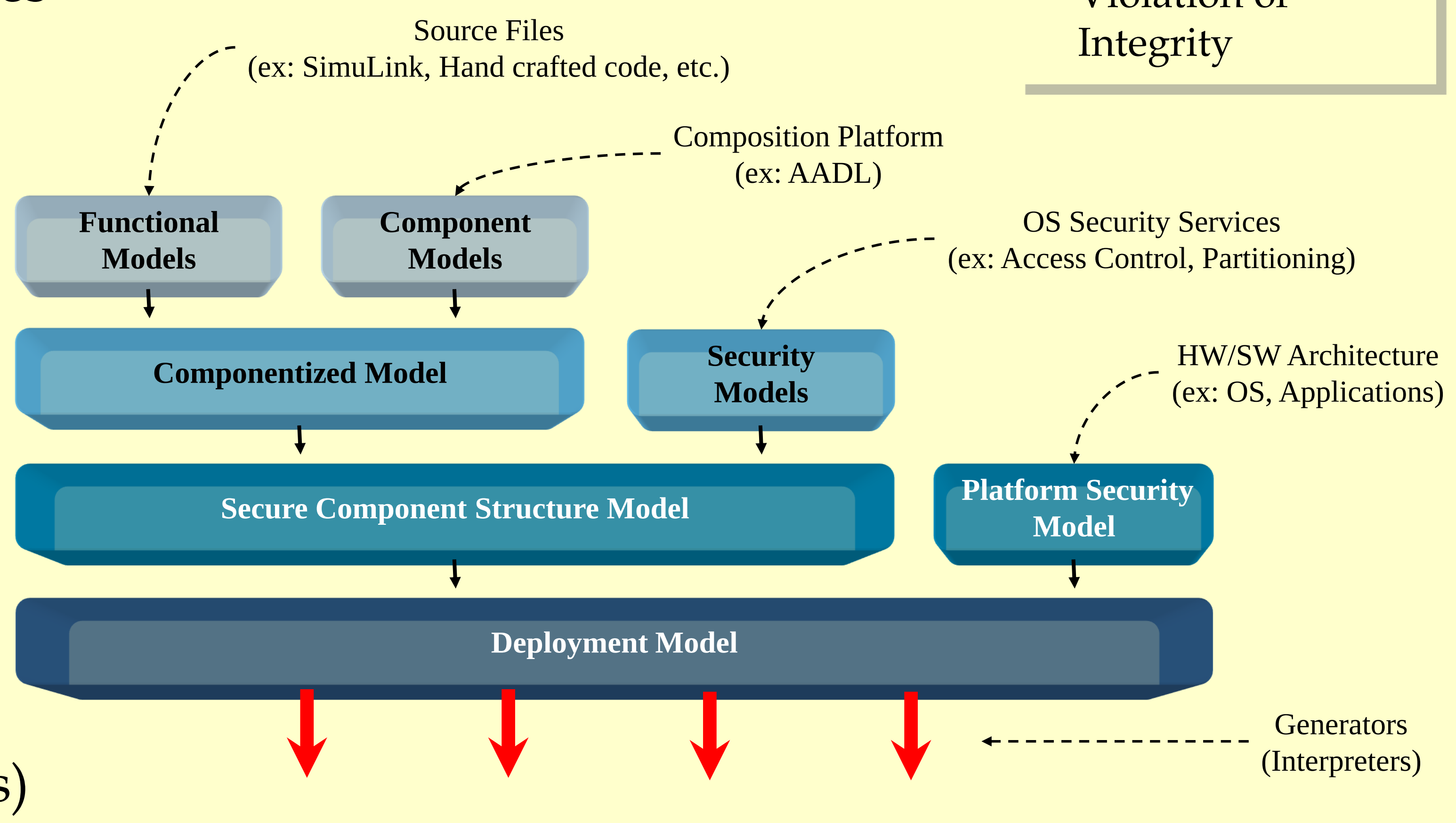
Security Extension examples

- Partitioning
- Role Based Access Control (RBAC)
- Secure Links
- Fair Exchange (guaranteed transactions)

Vulnerabilities

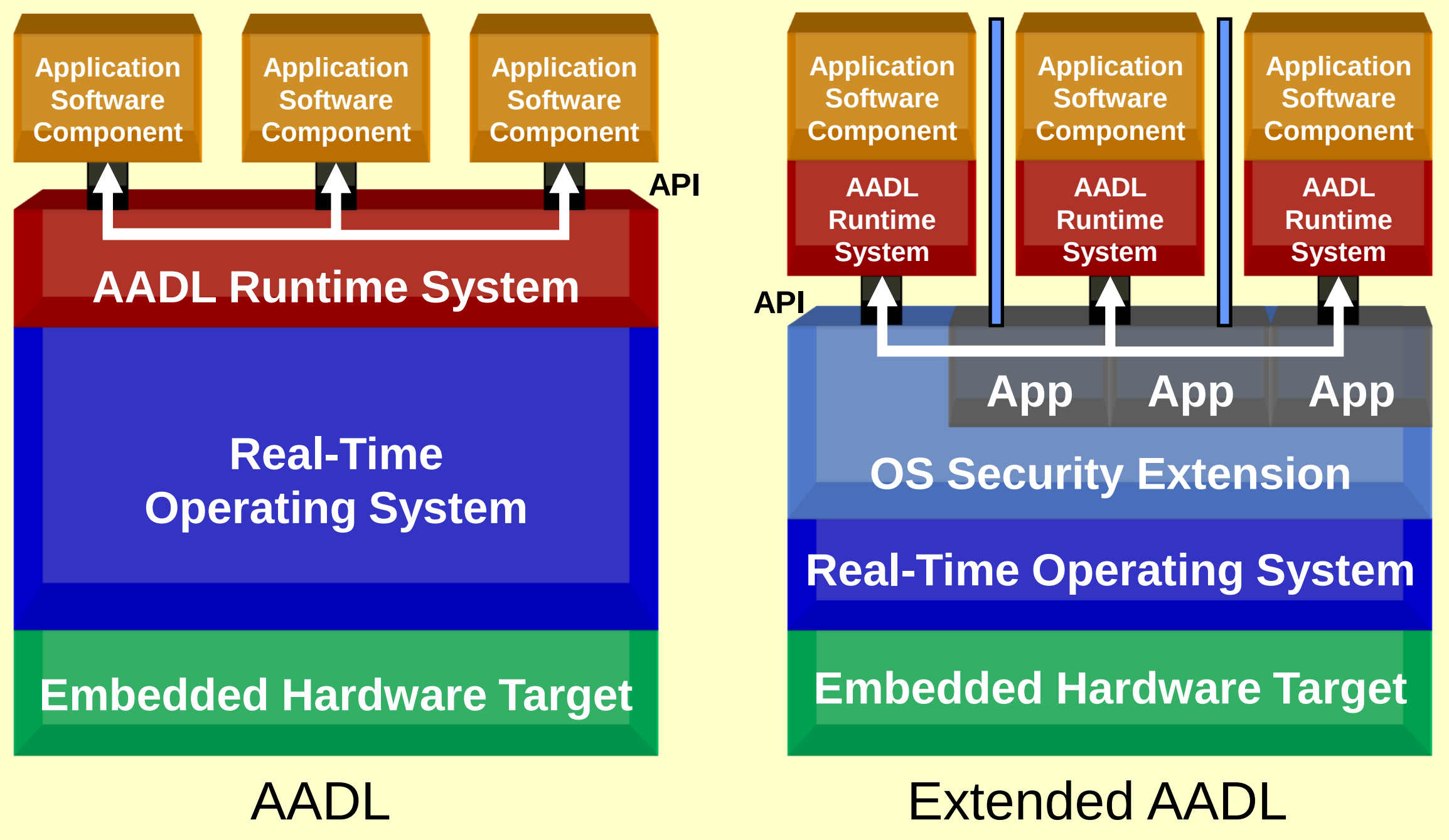
- Access Violation
- Violation of Autentication
- Violation of Confidentiality
- Violation of Integrity

Secure Code Generation



Comparison of AADL and the Security Extended AADL

AADL Execution Environment


Application Software Component
Application Software Component
Application Software Component
API
AADL Runtime System
Real-Time Operating System
Embedded Hardware Target
AADL
Application Software Component
Application Software Component
Application Software Component
AADL Runtime System
AADL Runtime System
AADL Runtime System
API
App
App
App
OS Security Extension
Real-Time Operating System
Embedded Hardware Target
Extended AADL

Gain with the Security Extended AADL

Introducing security at design level

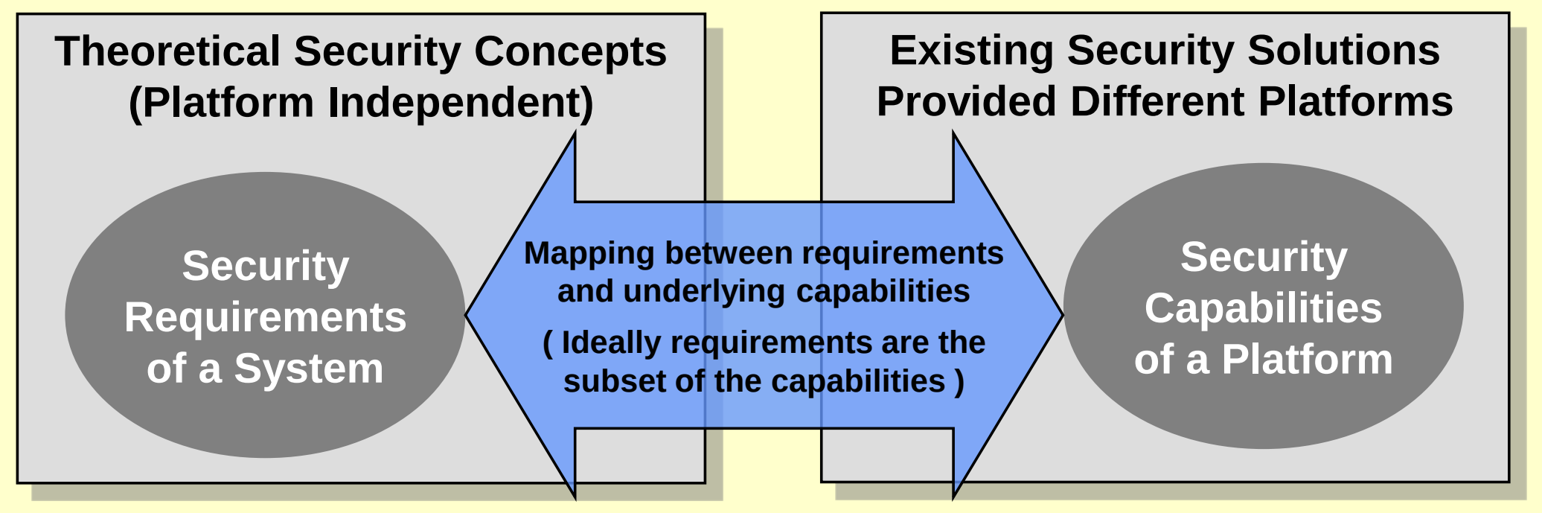
- Consistent and automatic configuration of security services offered by the operating system and middleware
- Investigating design tradeoffs between performance and security properties
- Verifying required security properties using explicit security models

Platform Security / Security Service Modeling

Abstracts out security properties of the platform that are essential for the design flow

Security Service Providers

- OS (ex: Linux, LynxOS, WinCE)
- HW (ex: Space Partitioning, Memory protection)
- Services of different applications
- (ex: Web Browser Based Authentication)


Theoretical Security Concepts
(Platform Independent)
Existing Security Solutions
Provided Different Platforms
Security Requirements of a System
Mapping between requirements and underlying capabilities
(Ideally requirements are the subset of the capabilities)
Security Capabilities of a Platform

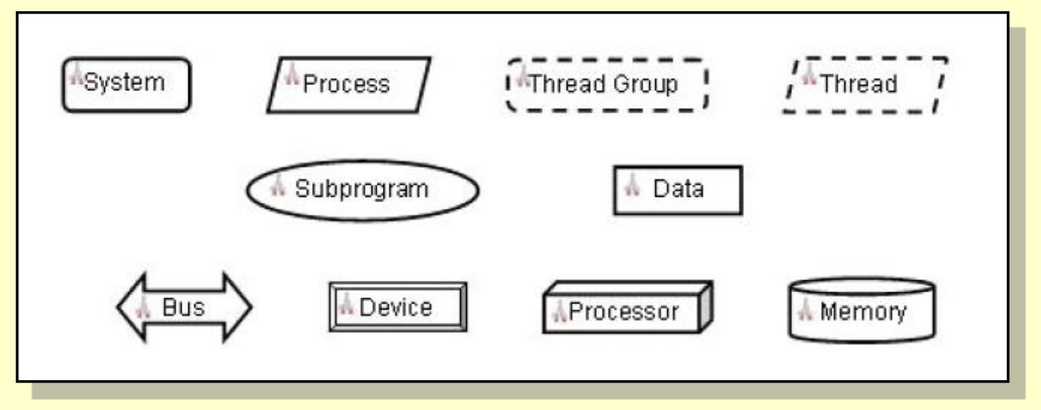
Platform Security Models with sufficient detail enable Code Generators to access Platform Specific Security Services

A DSML Example with the Security Extension

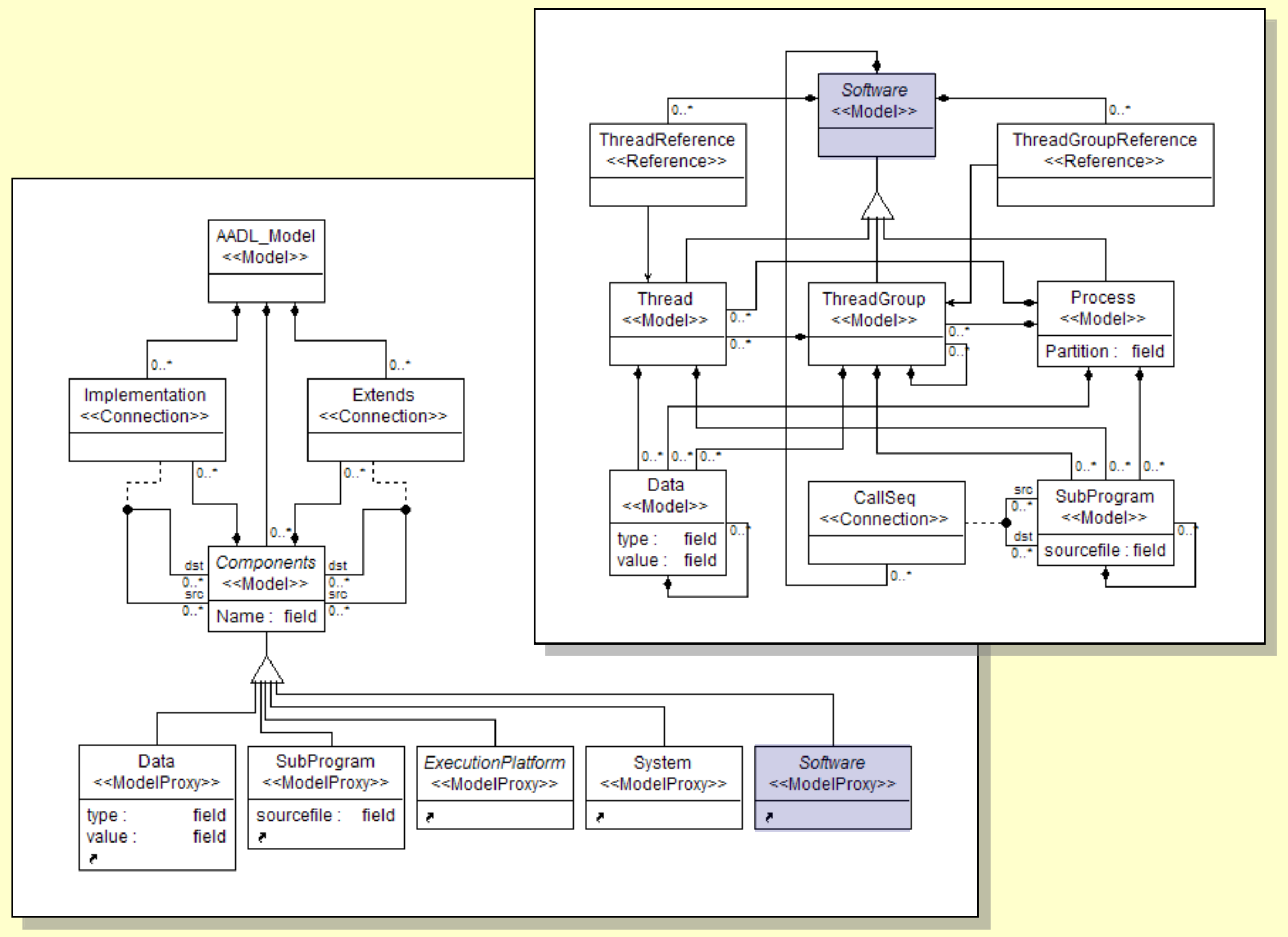
AADL with RBAC and Partitioning

Architectural Analysis and Design Language (AADL)

- Standard by SAE Aerospace (AS5506)
- Developed to model embedded systems with challenging resource constraints



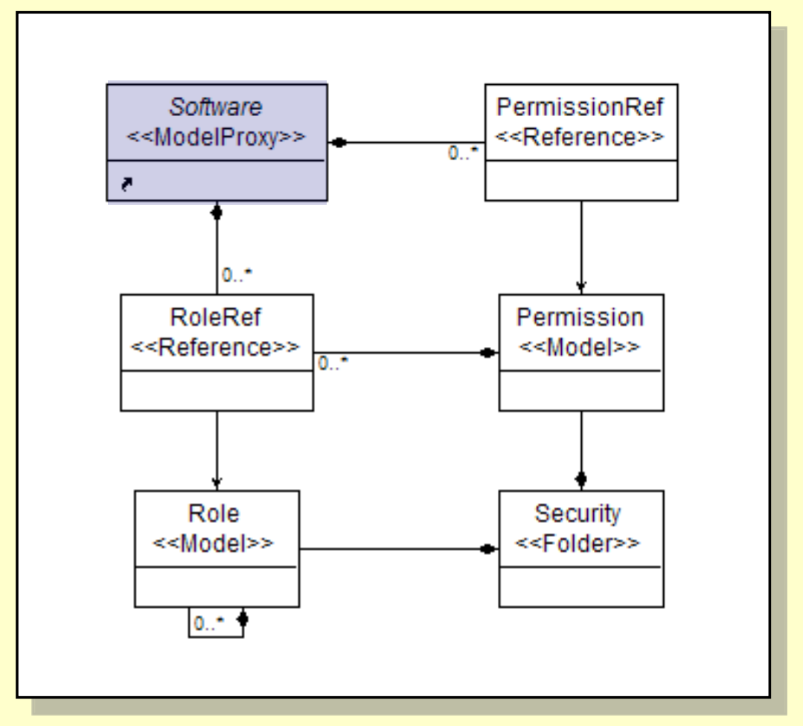
Definition of the AADL Language



Role Based Access Control (RBAC) building blocks

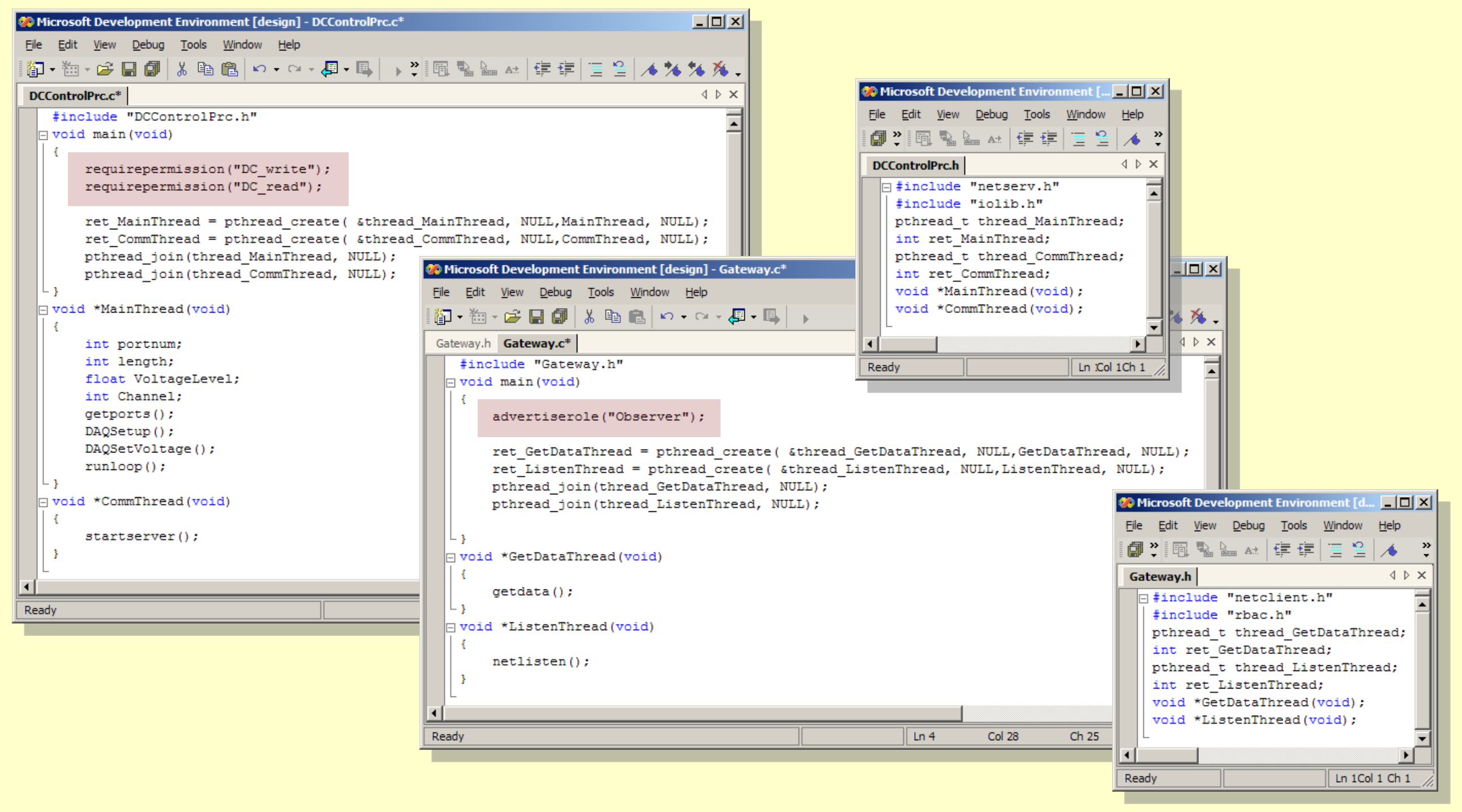
- Objects – subject to access control
- Operations – execution of some functions on objects
- Permissions – approval to perform operation on RBAC protected object
- Roles – job with assigned authority and responsibility
- Users – human being, machine, network or agent requesting operation on objects

Security Extension Metamodel



Automatic Code Generation

Code generators traverse the model and produce secure code that enforces the RBAC policies. The code generator makes use of the partitioning capabilities of the underlying platform.


Microsoft Development Environment [design] - DCControlPrc.c*
File
Edit
View
Debug
Tools
Window
Help
DCControlPrc.c*
Ready

```
#include "DCControlPrc.h"
void main(void)
{
    requirepermission("DC_write");
    requirepermission("DC_read");

    ret_MainThread = pthread_create( &thread_MainThread, NULL,MainThread, NULL);
    ret_CommThread = pthread_create( &thread_CommThread, NULL,CommThread, NULL);
    pthread_join(thread_MainThread, NULL);
    pthread_join(thread_CommThread, NULL);
}
void *MainThread(void)
{
    int portnum;
    int length;
    float VoltageLevel;
    int Channel;
    getports();
    DAQSetup();
    DAQSetVoltage();
    runloop();
}
void *CommThread(void)
{
    startserver();
}
```

```
#include "netserv.h"
#include "iolib.h"
pthread_t thread_MainThread;
int ret_MainThread;
pthread_t thread_CommThread;
int ret_CommThread;
void *MainThread(void);
void *CommThread(void);
```

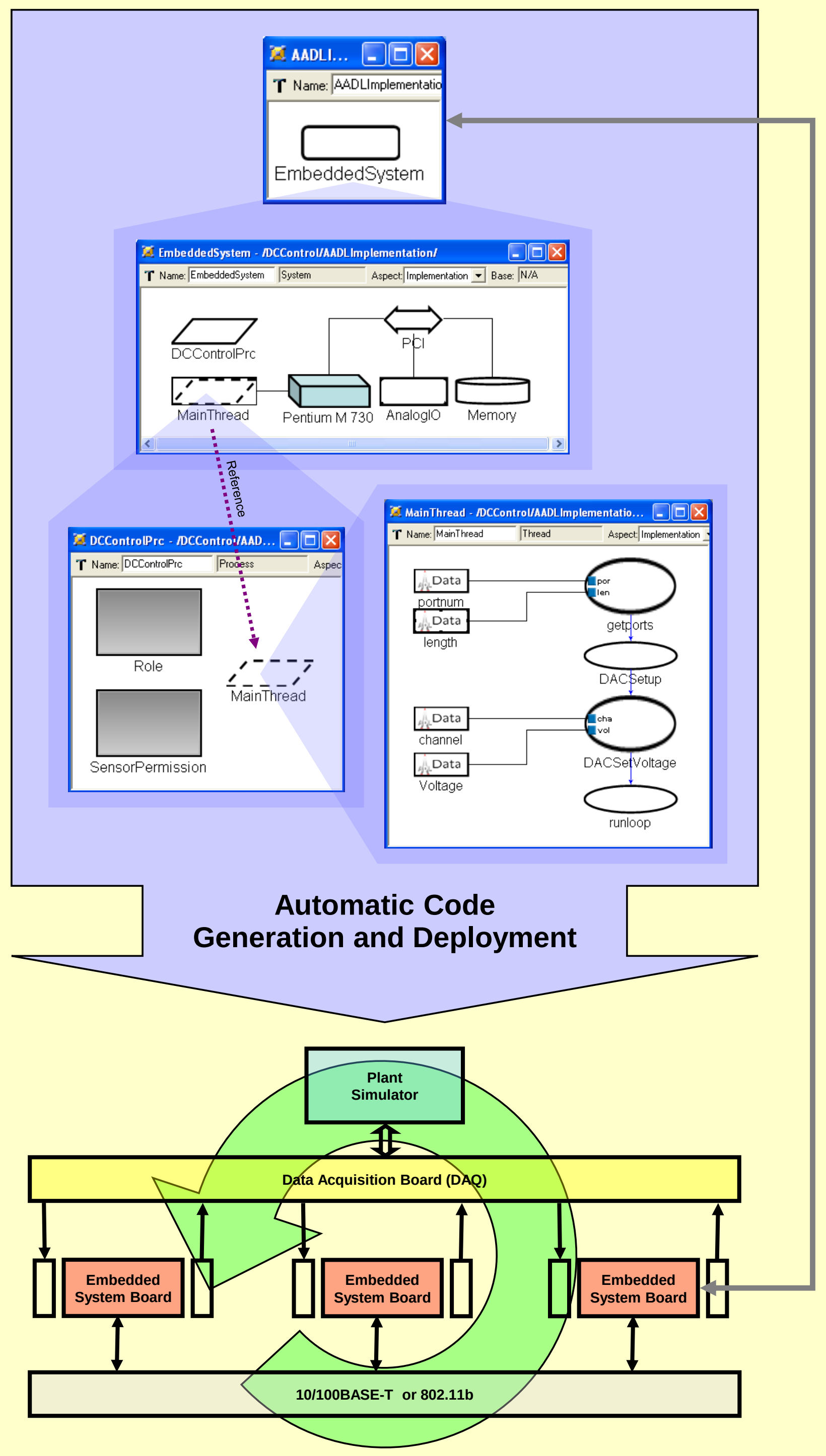
```
#include "Gateway.h"
void main(void)
{
    advertiserole("Observer");

    ret_GetDataThread = pthread_create( &thread_GetDataThread, NULL,GetDataThread, NULL);
    ret_ListenThread = pthread_create( &thread_ListenThread, NULL,ListenThread, NULL);
    pthread_join(thread_GetDataThread, NULL);
    pthread_join(thread_ListenThread, NULL);

}
void *GetDataThread(void)
{
    getdata();
}
void *ListenThread(void)
{
    netlisten();
}
```

```
#include "netclient.h"
#include "rbac.h"
pthread_t thread_GetDataThread;
int ret_GetDataThread;
pthread_t thread_ListenThread;
int ret_ListenThread;
void *GetDataThread(void);
void *ListenThread(void);
```

The process of AADL code generation



April 27-28, 2006