STAC: Software Tuning Panels for Autonomic Control

Automatically Identifying Software Tuning Parameters

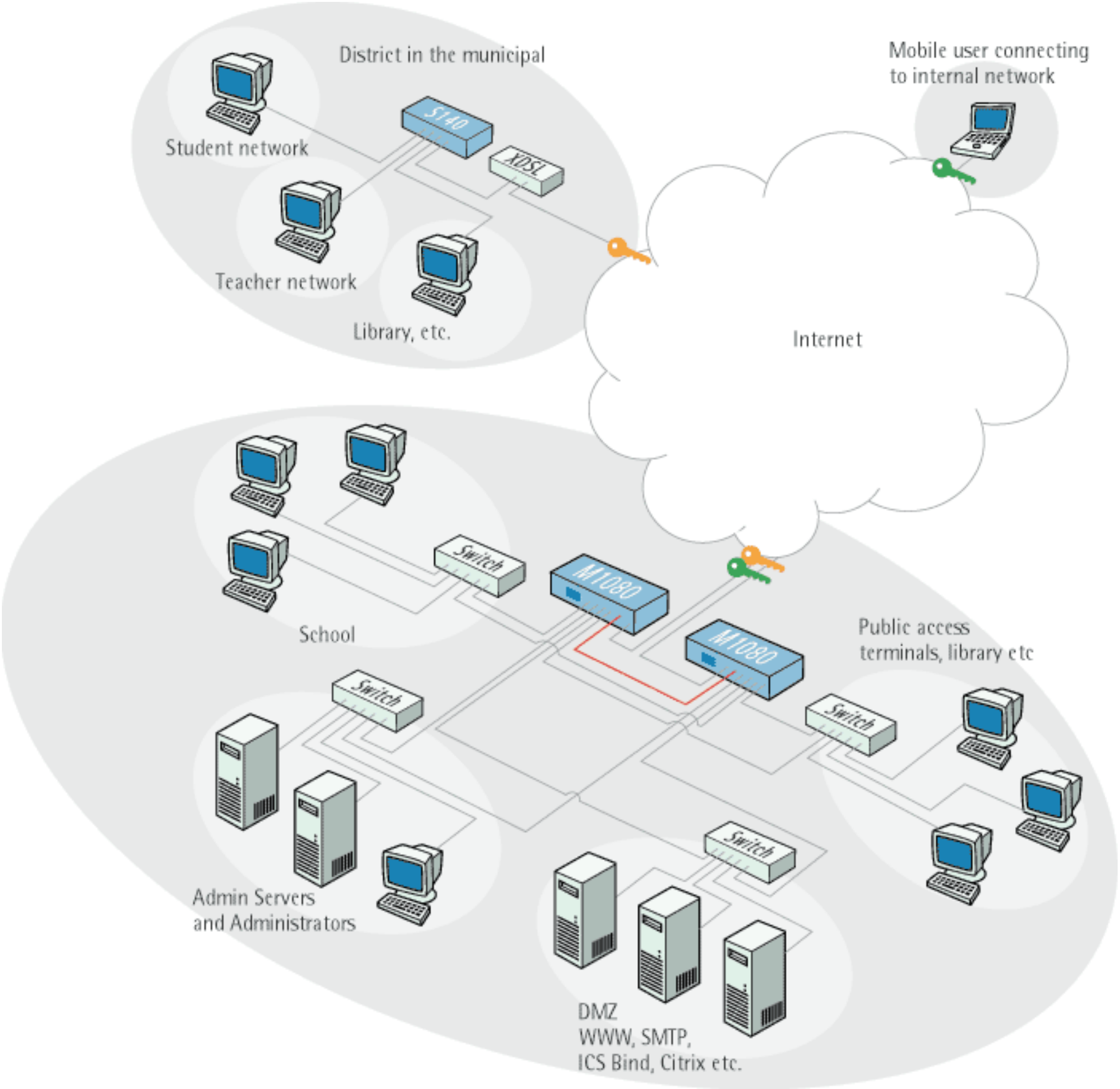
Project Participants

- Nevon Brake, Queen’s University
- Jim Cordy, Queen’s University
- Elizabeth Dancy, IBM
- Marin Litoiu, IBM
- Valentina Popescu, IBM

“Computing systems’ complexity appears to be approaching the limits of human capability...”

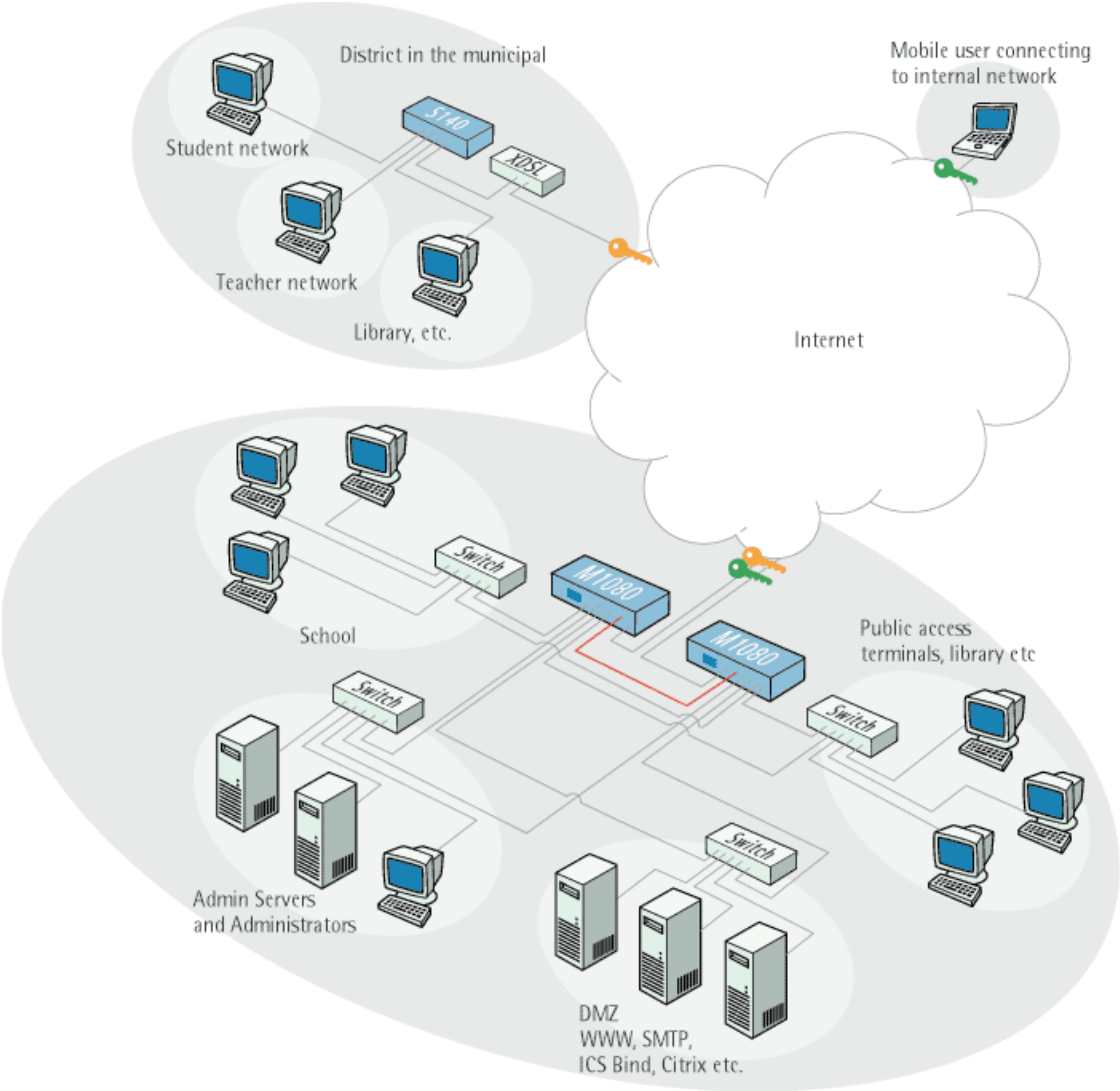
The Vision of Autonomic Computing, 2003

Pervasive Computing


District in the municipal
Student network
S140
XDSL
Teacher network
Library, etc.
Mobile user connecting to internal network
Internet
School
Switch
M1080
M1080
Public access terminals, library etc
Switch
Switch
Admin Servers and Administrators
Switch
DMZ WWW, SMTP, ICS Bind, Citrix etc.

Pervasive Computing

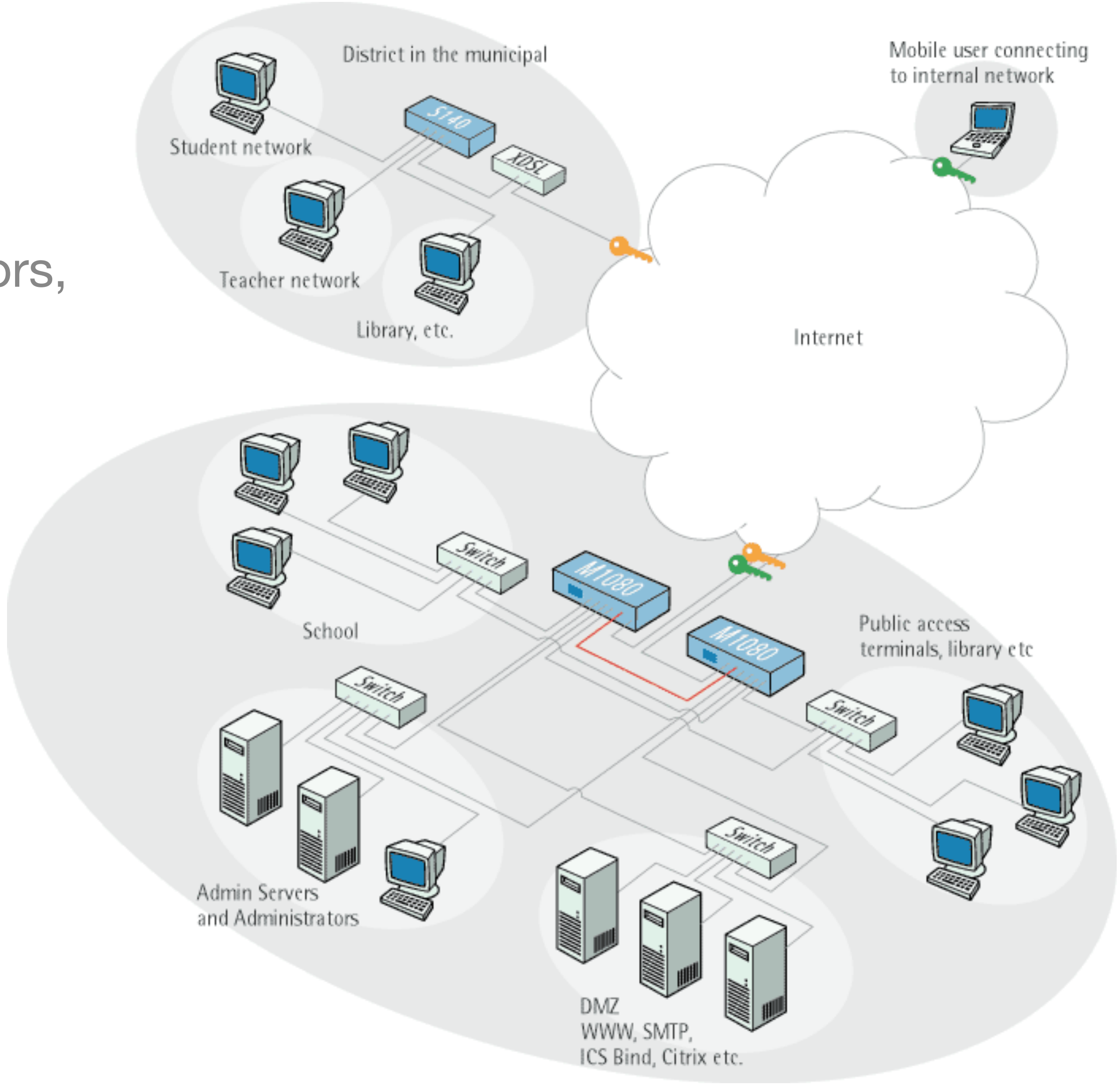
- Interconnected (i.e., complex dependencies)


District in the municipal
S140
XDSL
Student network
Teacher network
Library, etc.
Mobile user connecting to internal network
Internet
Switch
M1080
M1080
School
Public access terminals, library etc
Switch
Switch
Switch
Admin Servers and Administrators
DMZ
WWW, SMTP,
ICS Bind, Citrix etc.

Pervasive Computing

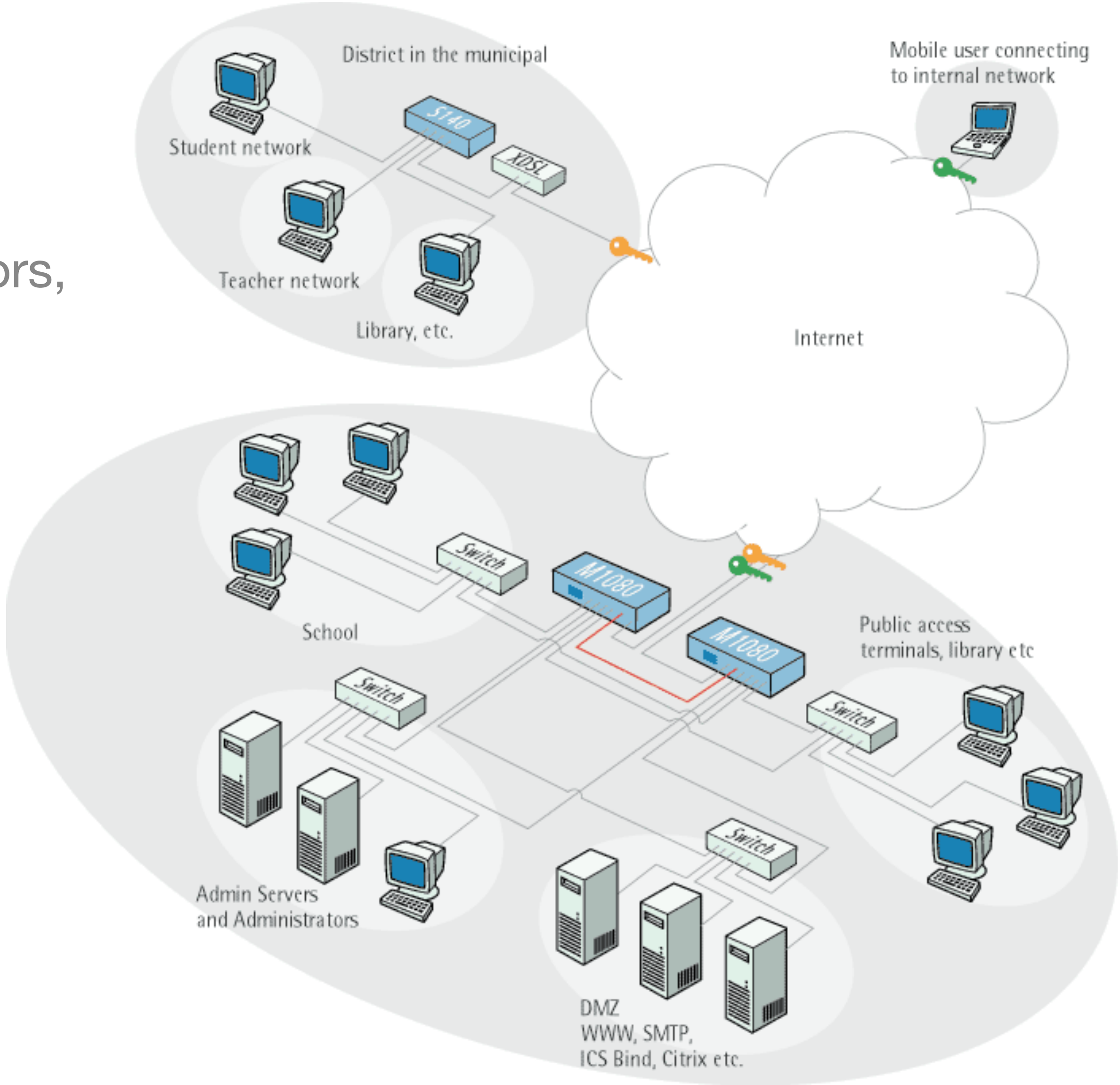
- Interconnected (i.e., complex dependencies)

- Diverse (e.g., hardware vendors, software versions)


District in the municipal
Student network
S140
XDSL
Teacher network
Library, etc.
Mobile user connecting to internal network
Internet
School
Switch
M1080
M1080
Public access terminals, library etc
Switch
Admin Servers and Administrators
Switch
DMZ
WWW, SMTP,
ICS Bind, Citrix etc.

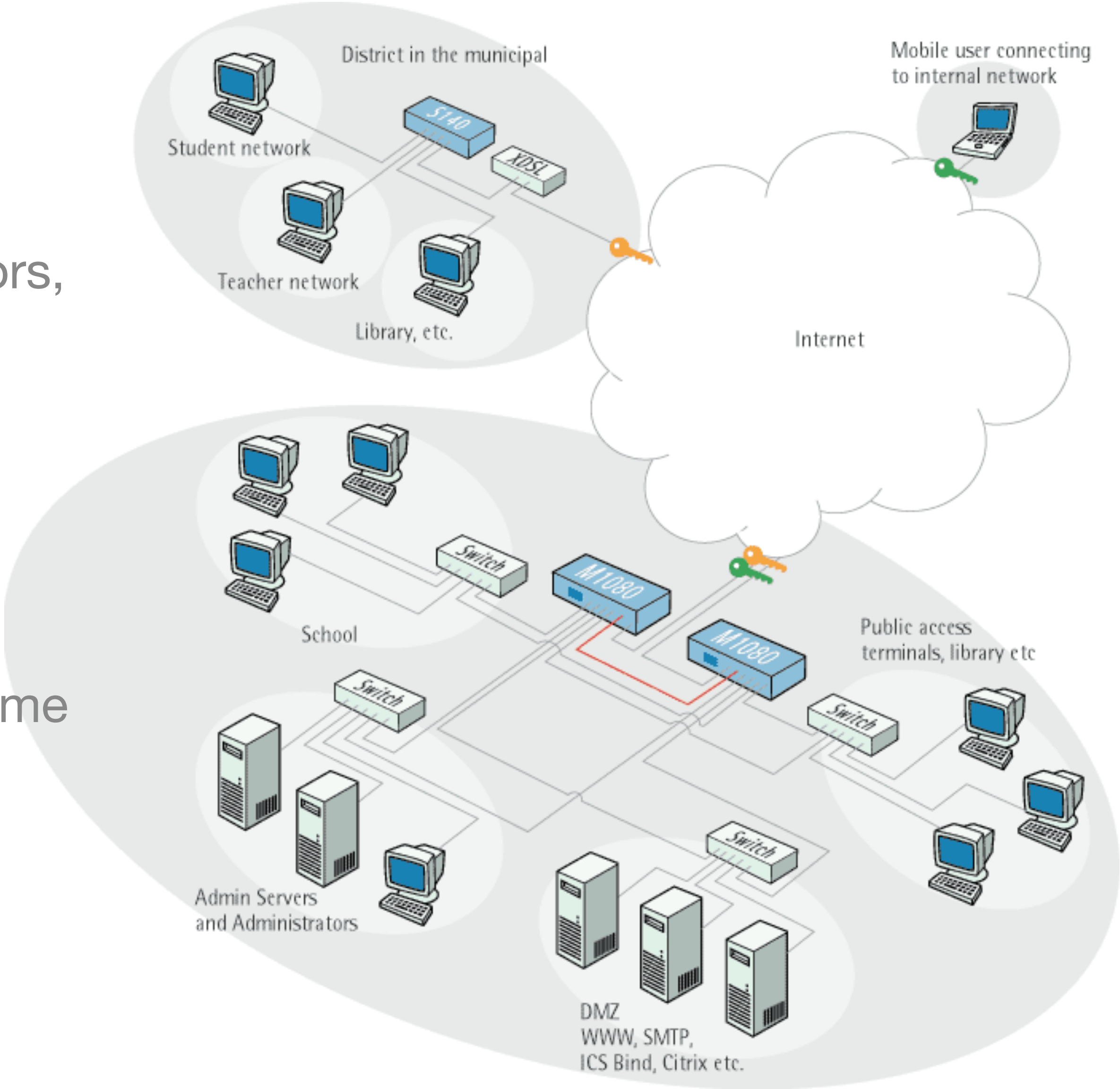
Pervasive Computing

- Interconnected (i.e., complex dependencies)
- Diverse (e.g., hardware vendors, software versions)
- Unanticipated interactions

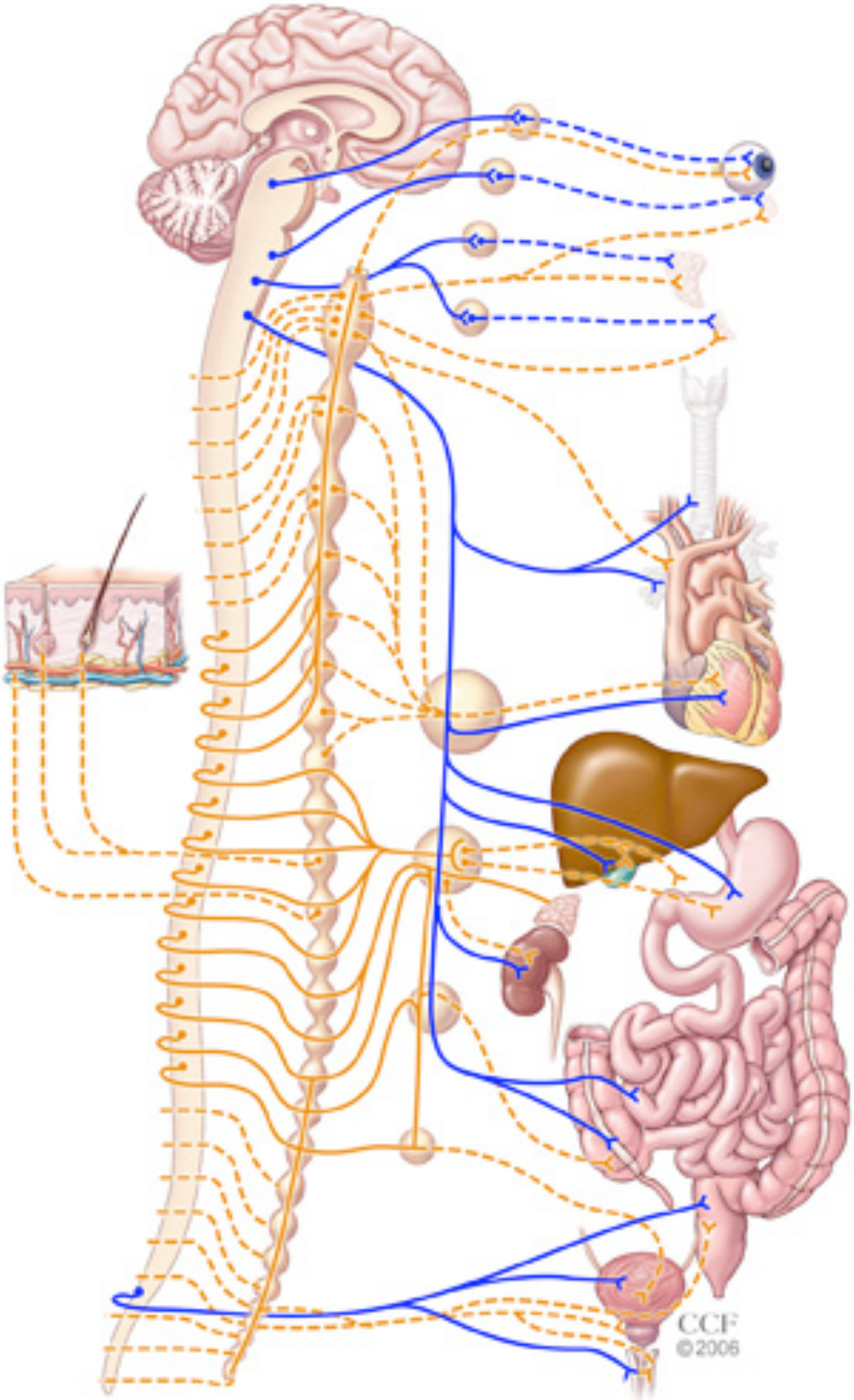

District in the municipal
Mobile user connecting to internal network
Student network
S140
XDSL
Teacher network
Library, etc.
Internet
Switch
M1080
M1080
School
Public access terminals, library etc
Switch
Switch
Admin Servers and Administrators
Switch
DMZ WWW, SMTP, ICS Bind, Citrix etc.

Pervasive Computing

- Interconnected (i.e., complex dependencies)
- Diverse (e.g., hardware vendors, software versions)
- Unanticipated interactions
- Runtime monitoring and diagnosis (e.g., 99.999% uptime requirements)

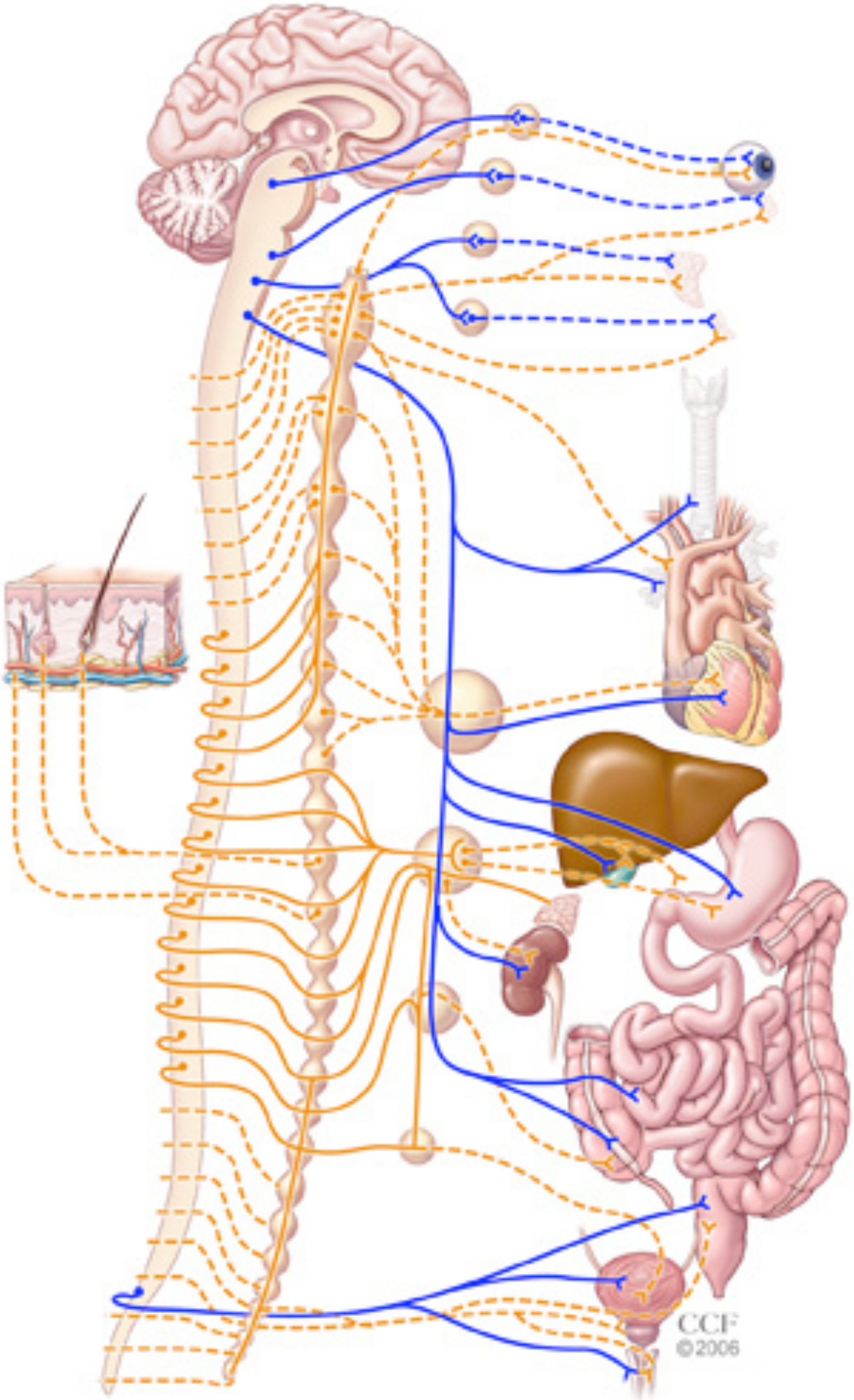

District in the municipal
S140
XDSL
Student network
Teacher network
Library, etc.
Mobile user connecting to internal network
Internet
Switch
M1080
M1080
School
Public access terminals, library etc
Switch
Switch
Admin Servers and Administrators
Switch
DMZ WWW, SMTP, ICS Bind, Citrix etc.

The Autonomic Vision



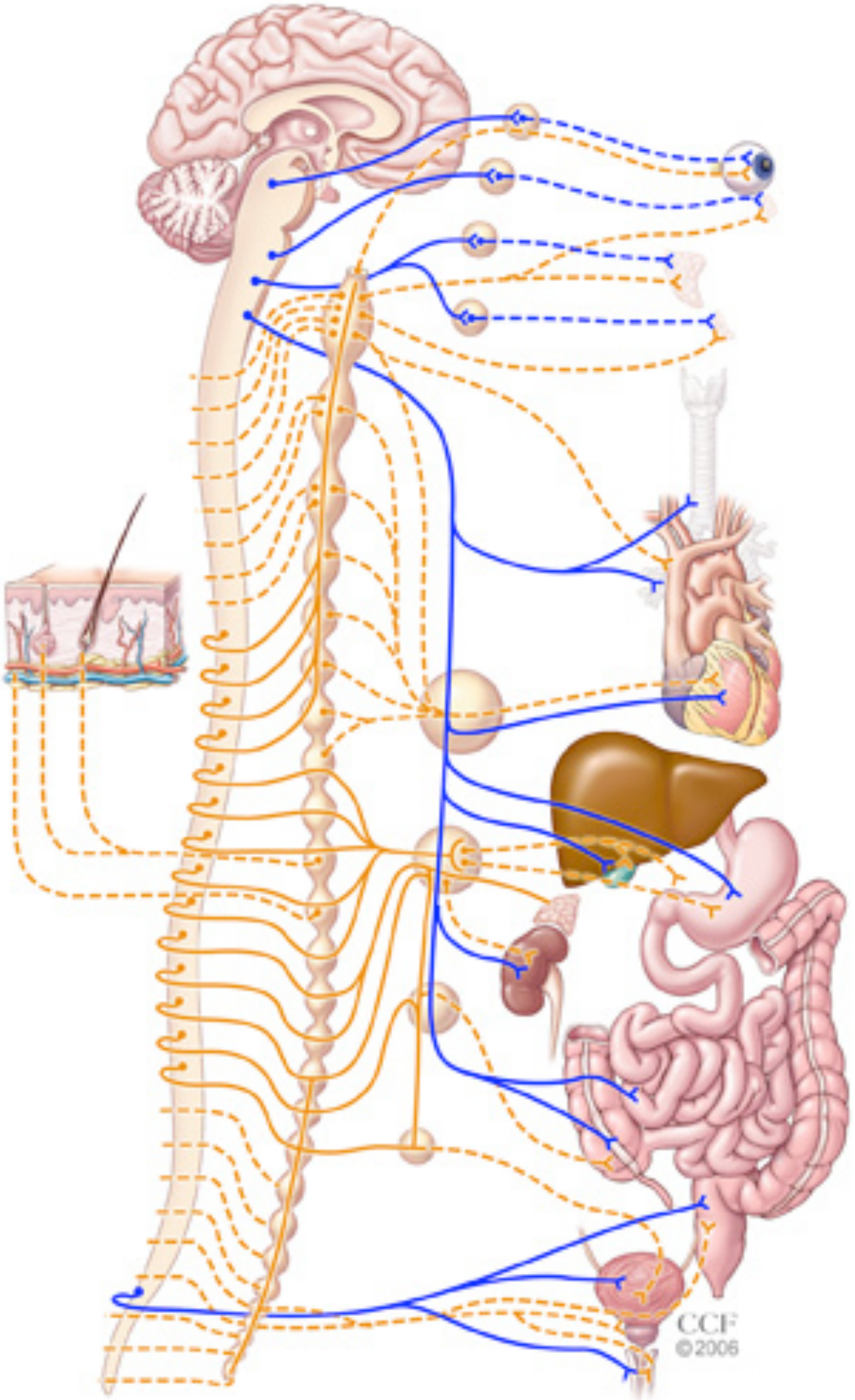
The Autonomic Vision

- Self-configuration



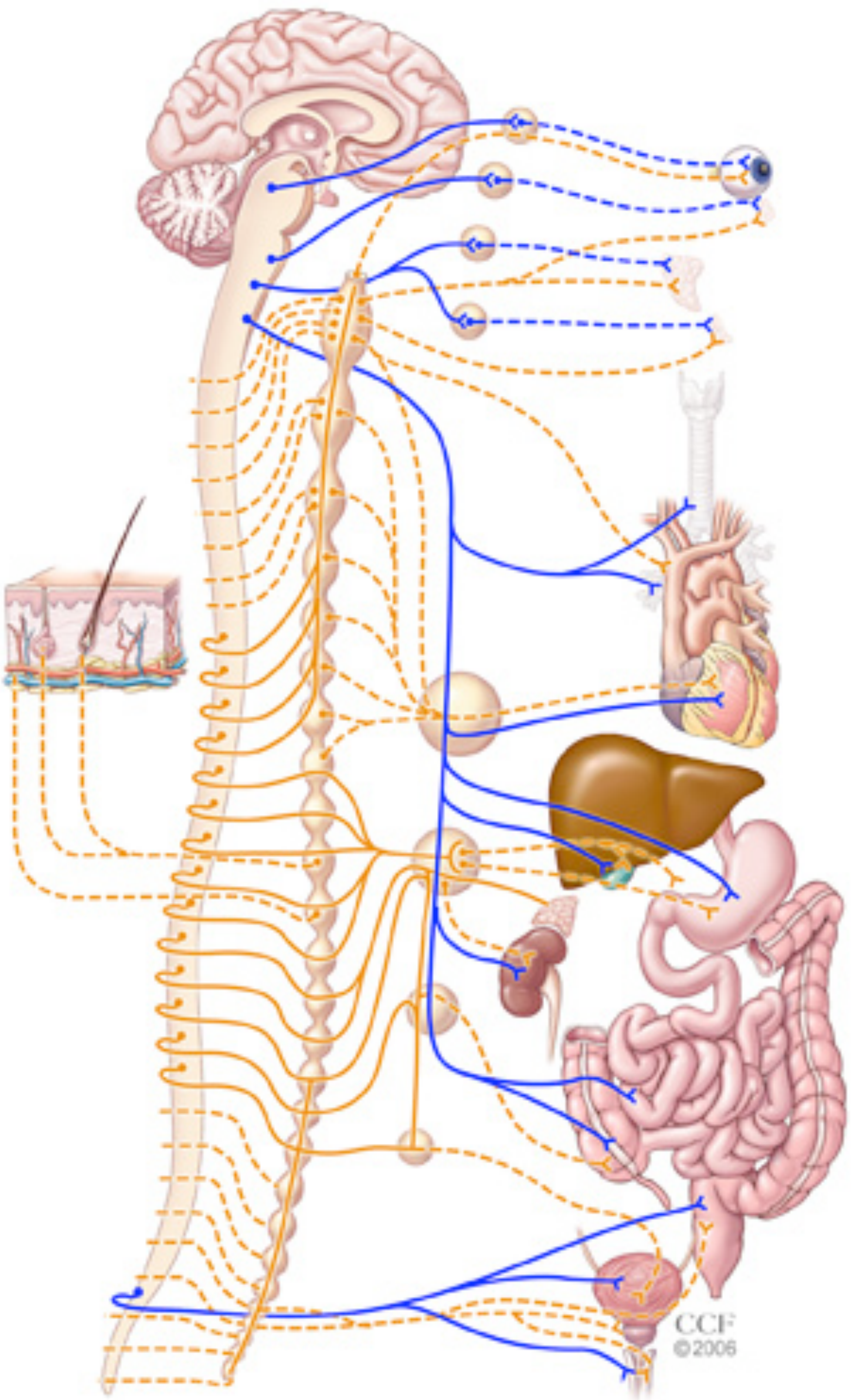
The Autonomic Vision

- Self-configuration
- Self-optimization



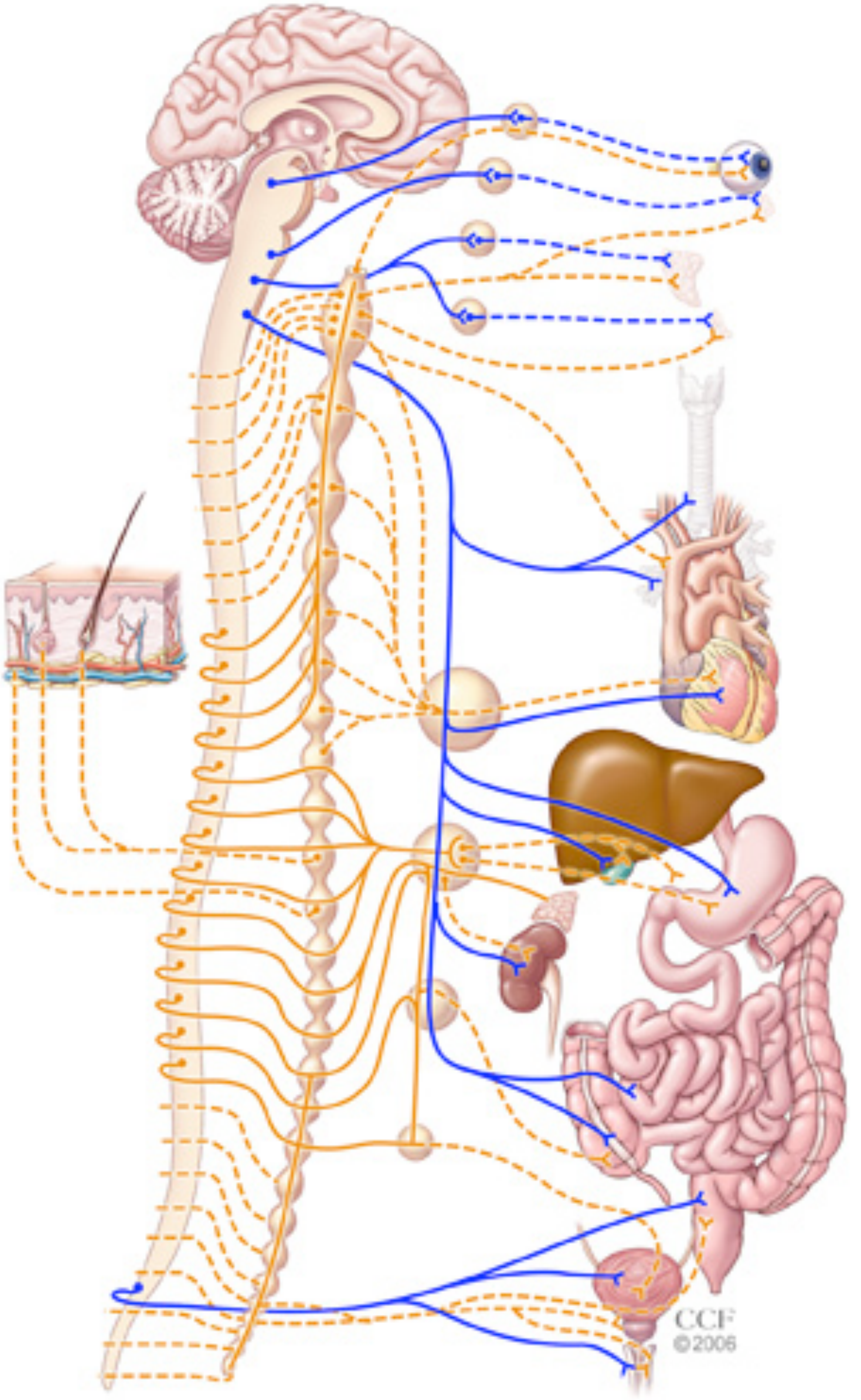
The Autonomic Vision

- Self-configuration
- Self-optimization
- Self-healing



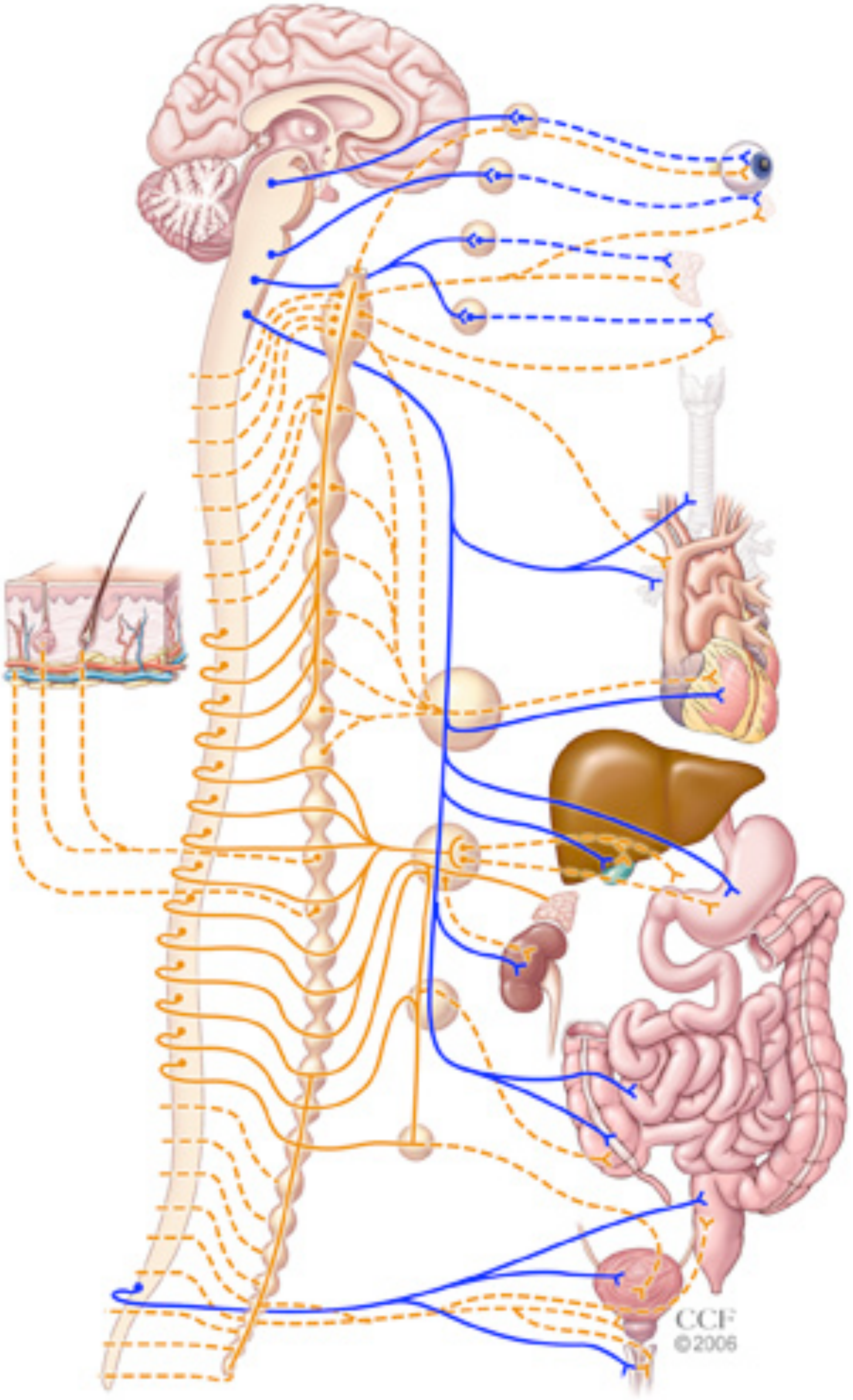
The Autonomic Vision

- Self-configuration
- Self-optimization
- Self-healing
- Self-protection

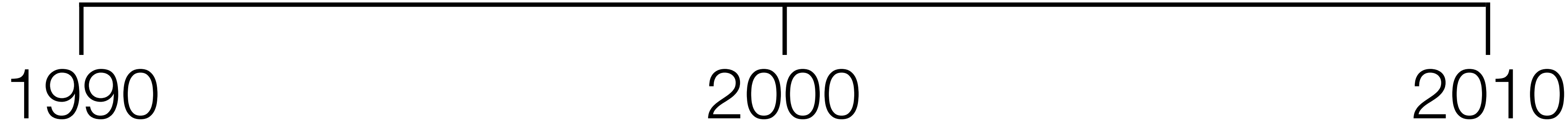


The Autonomic Vision

- Self-configuration
- Self-optimization
- Self-healing
- Self-protection
- Requires “Self-” awareness!



New Thinking, Old Problem!

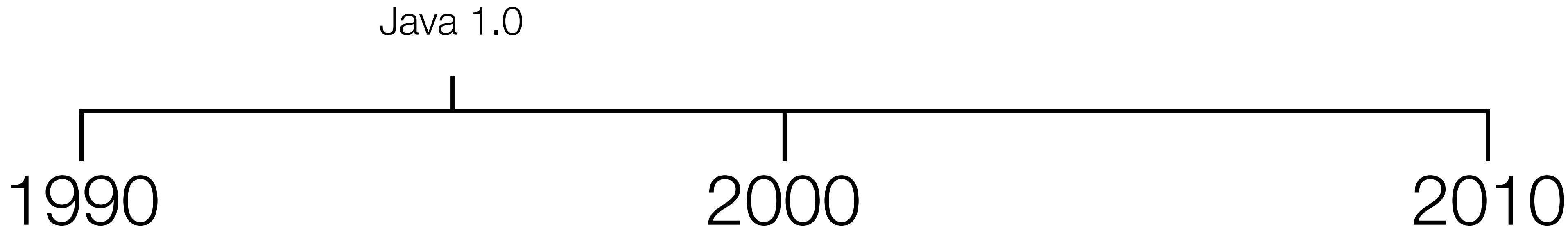


1990

2000

2010

New Thinking, Old Problem!



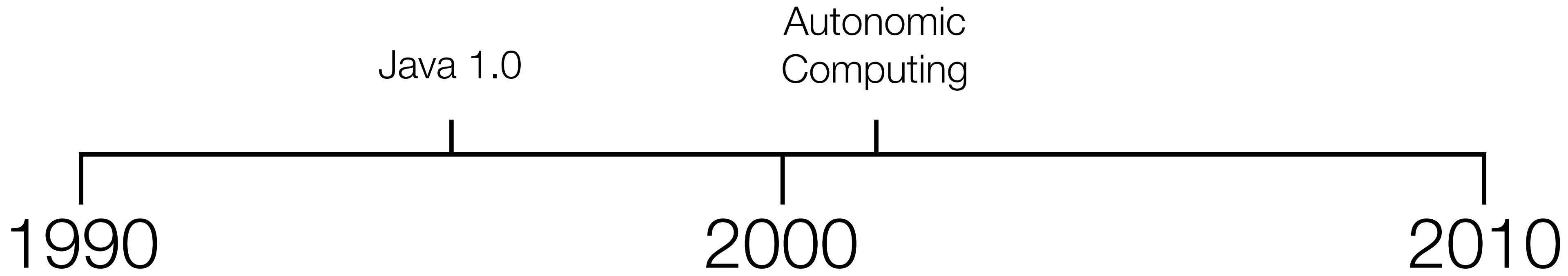
Java 1.0

1990

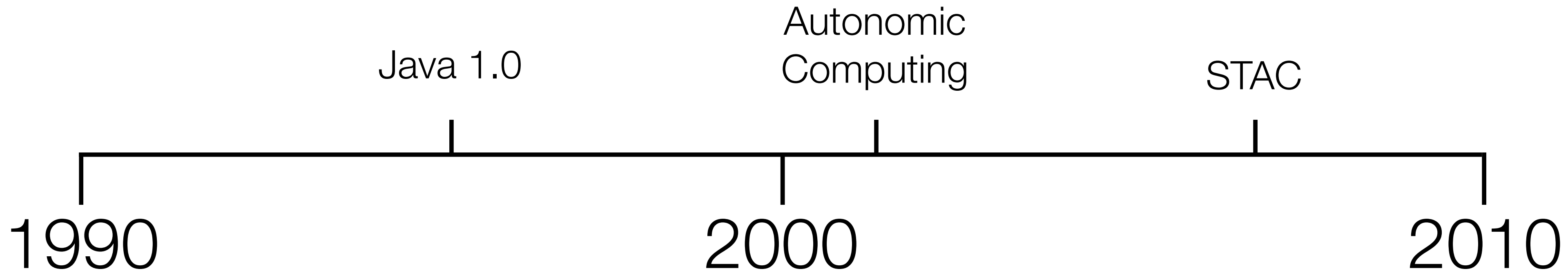
2000

2010

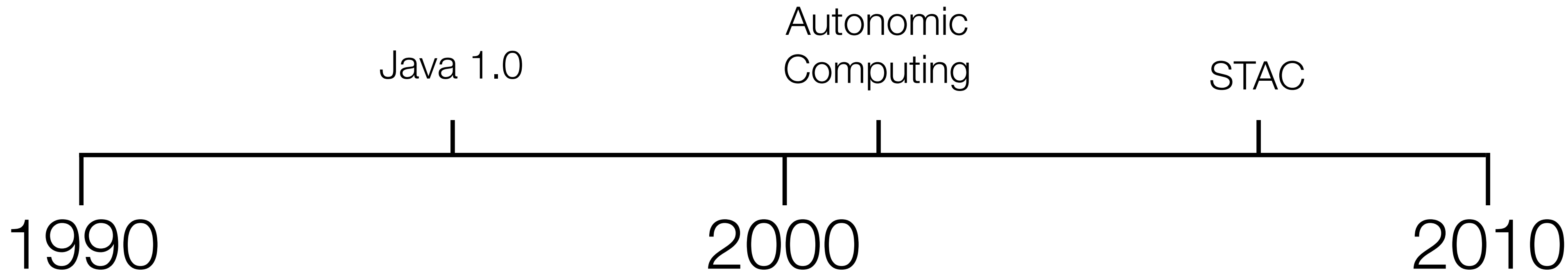
New Thinking, Old Problem!



New Thinking, Old Problem!

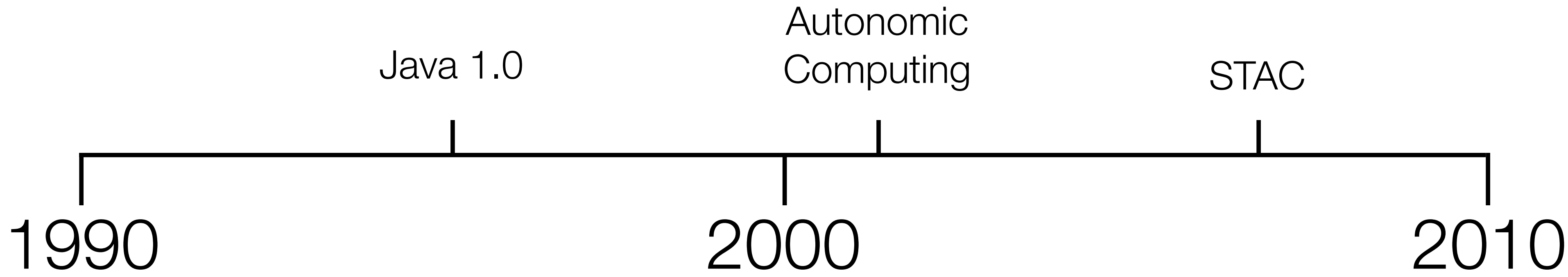


New Thinking, Old Problem!



- 400 features in 36 distinct IBM products have autonomic computing capabilities [7]

New Thinking, Old Problem!



- 400 features in 36 distinct IBM products have autonomic computing capabilities [7]
- Design vs. evolution for an autonomic future

Challenges for Software Evolution

Challenges for Software Evolution

- Manual rearchitecture is impractical

Challenges for Software Evolution

- Manual rearchitecture is impractical
- Analysis and rearchitecture requires tool support

Challenges for Software Evolution

- Manual rearchitecture is impractical
- Analysis and rearchitecture requires tool support
- Semantics must be preserved

Challenges for Software Evolution

- Manual rearchitecture is impractical
- Analysis and rearchitecture requires tool support
- Semantics must be preserved
- Syntax must be maintainable

STAC in a Nutshell

STAC in a Nutshell

- Automates the analysis and rearchitecture of existing systems for autonomic control by identifying, exposing and isolating tuning parameters

STAC in a Nutshell

- Automates the analysis and rearchitecture of existing systems for autonomic control by identifying, exposing and isolating tuning parameters

- Tuning parameters are observed and controlled from a common control panel

STAC in a Nutshell

- Automates the analysis and rearchitecture of existing systems for autonomic control by identifying, exposing and isolating tuning parameters

- Tuning parameters are observed and controlled from a common control panel

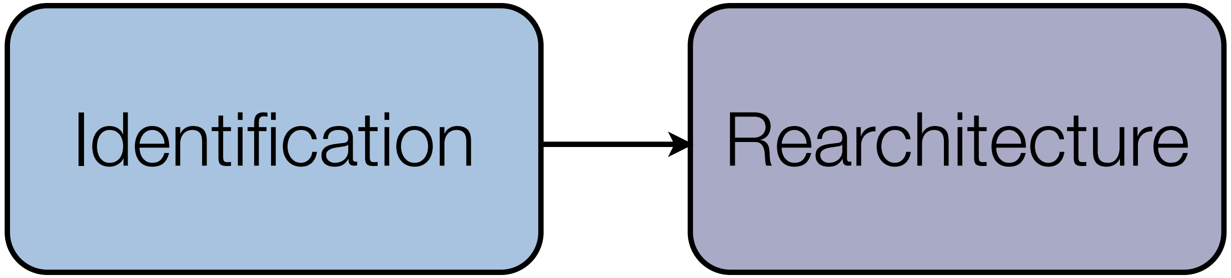
Tuning Parameter: A scalar field, or scalar property of a structured field, explicitly declared in source code that is used to observe or influence the behaviour of a system in terms of metrics like performance or security.

The STAC Pipeline: Past, Present and Future

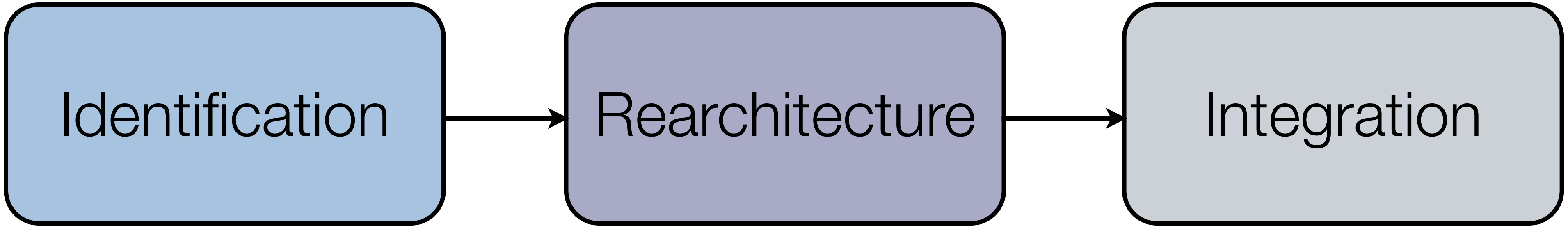
The STAC Pipeline: Past, Present and Future

Rearchitecture

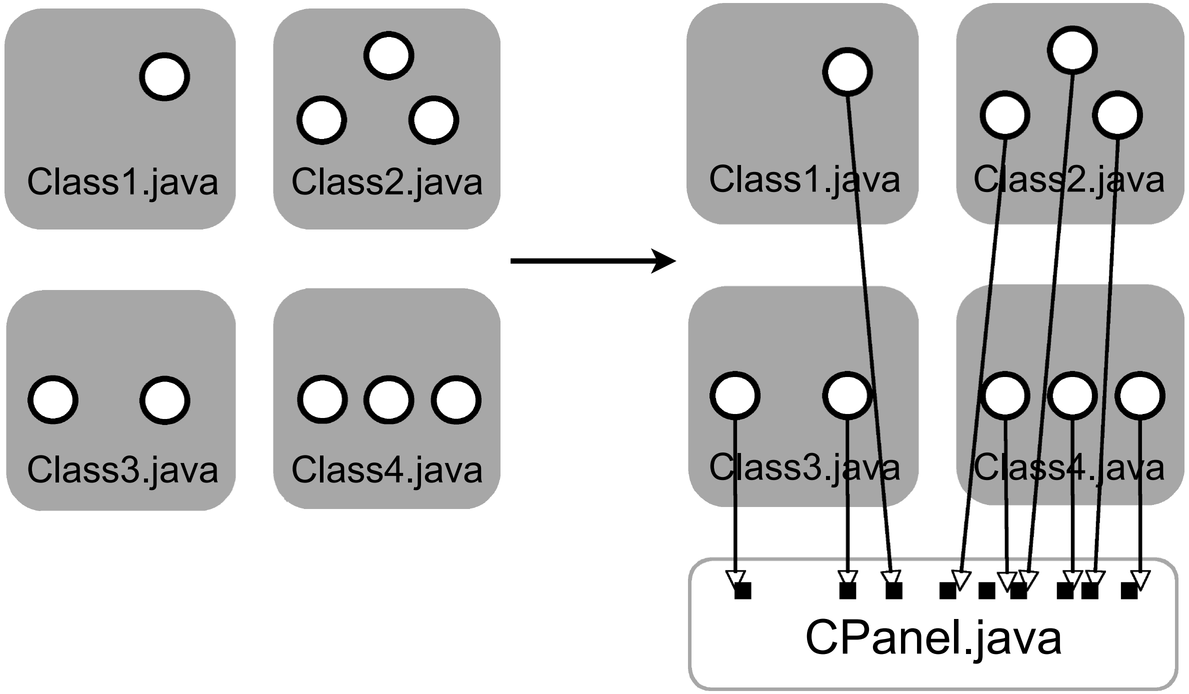
The STAC Pipeline: Past, Present and Future



The STAC Pipeline: Past, Present and Future

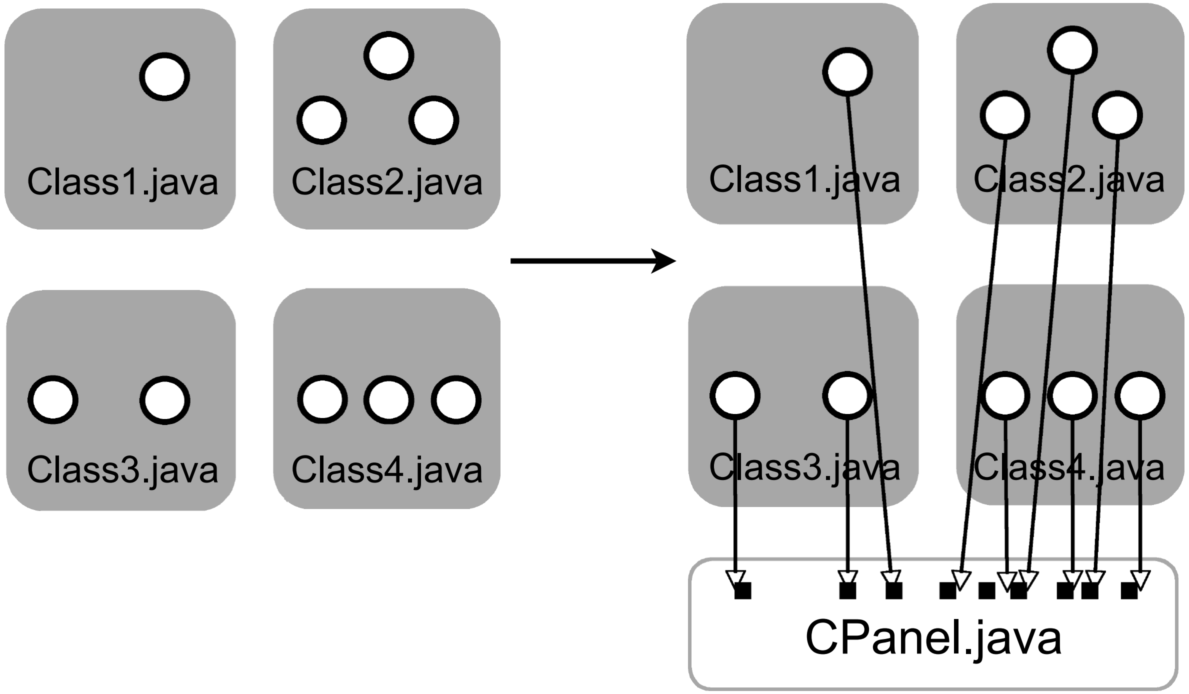


Rearchitecture using TXL (Past)



Rearchitecture using TXL (Past)

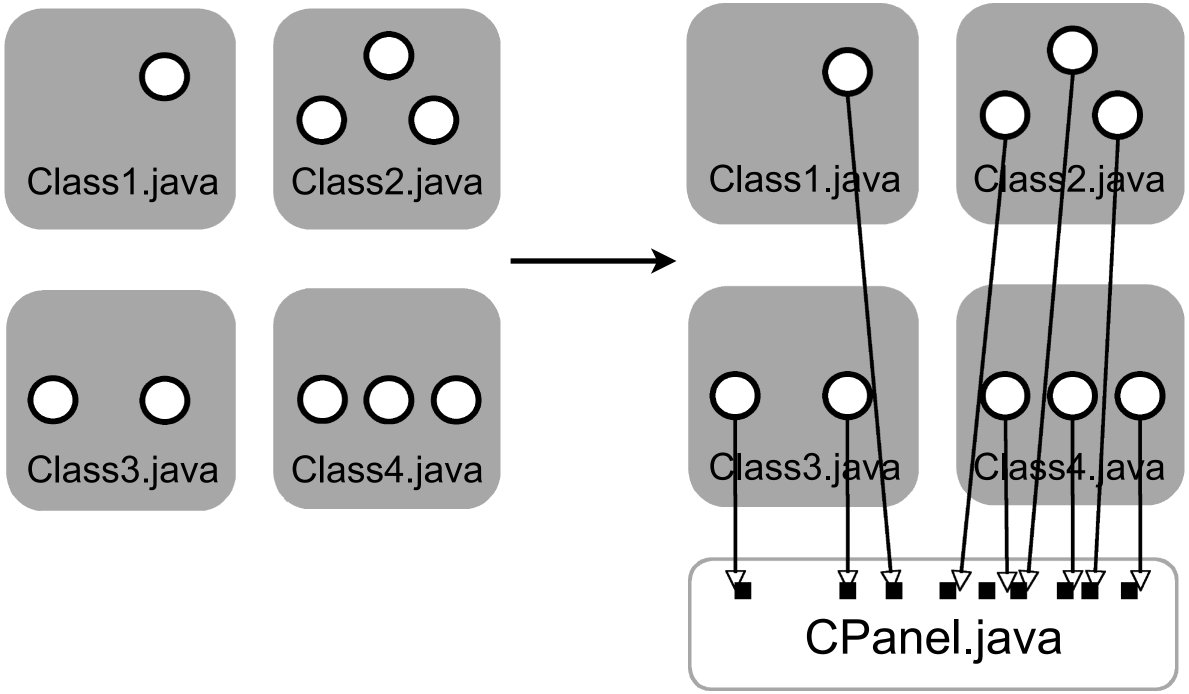
- Markup declarations with XML



Rearchitecture using TXL (Past)

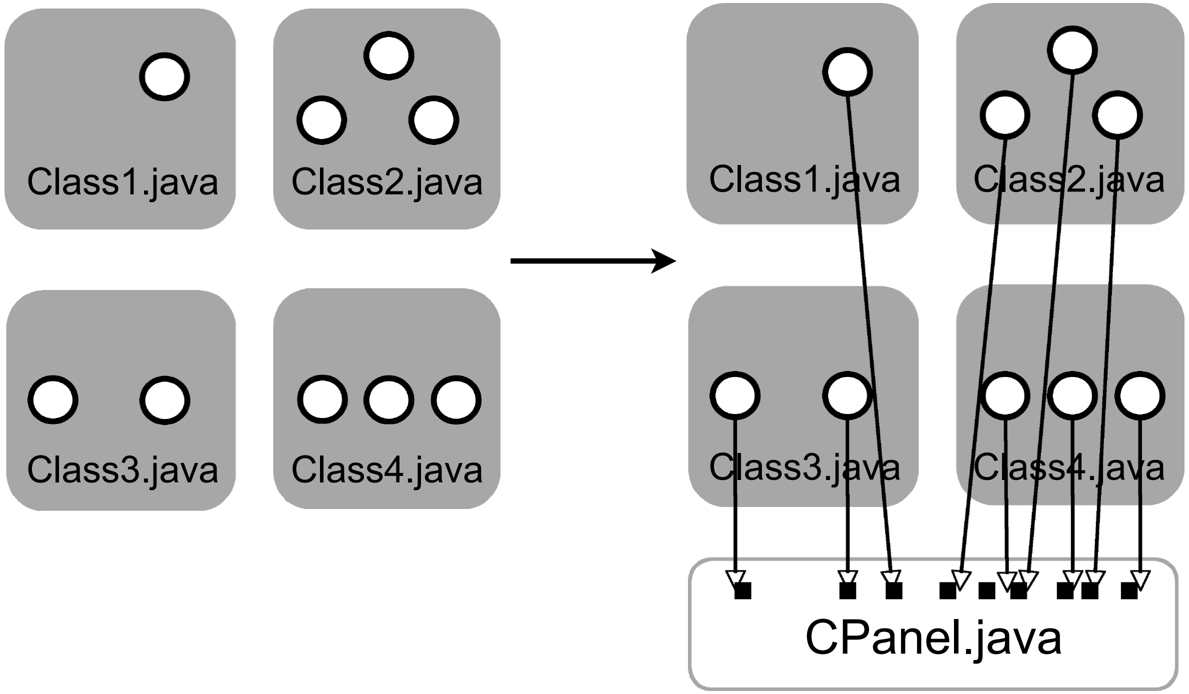
- Markup declarations with XML

- Normalize source files



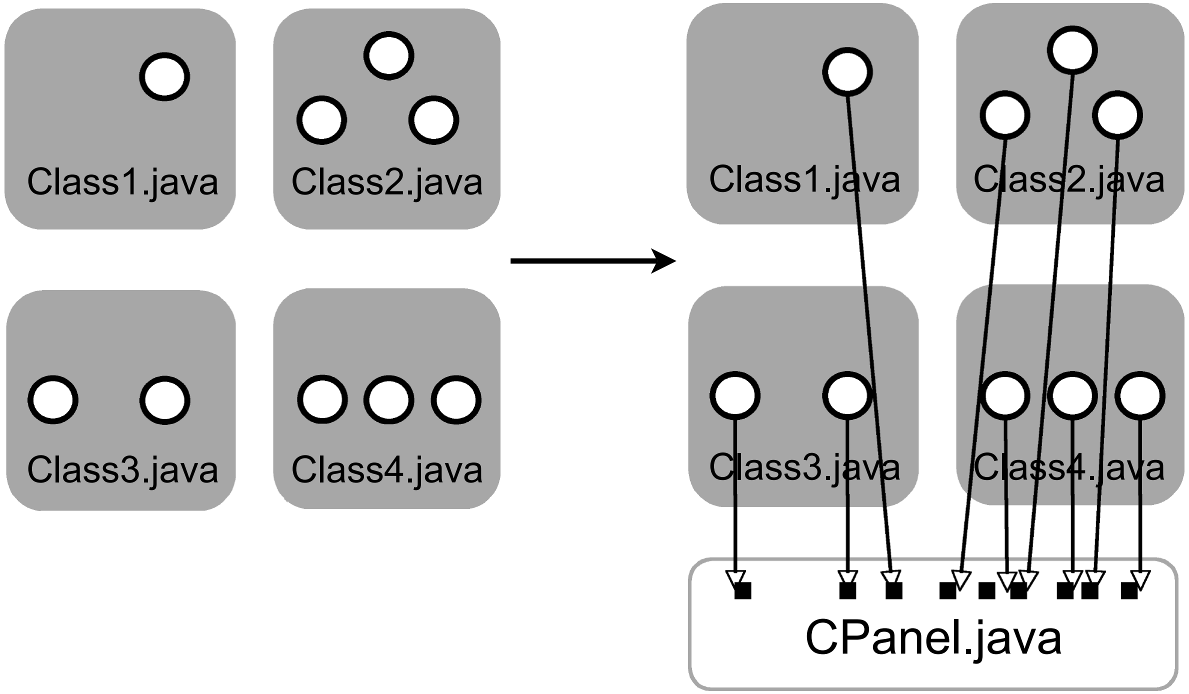
Rearchitecture using TXL (Past)

- Markup declarations with XML
- Normalize source files
- Trace references



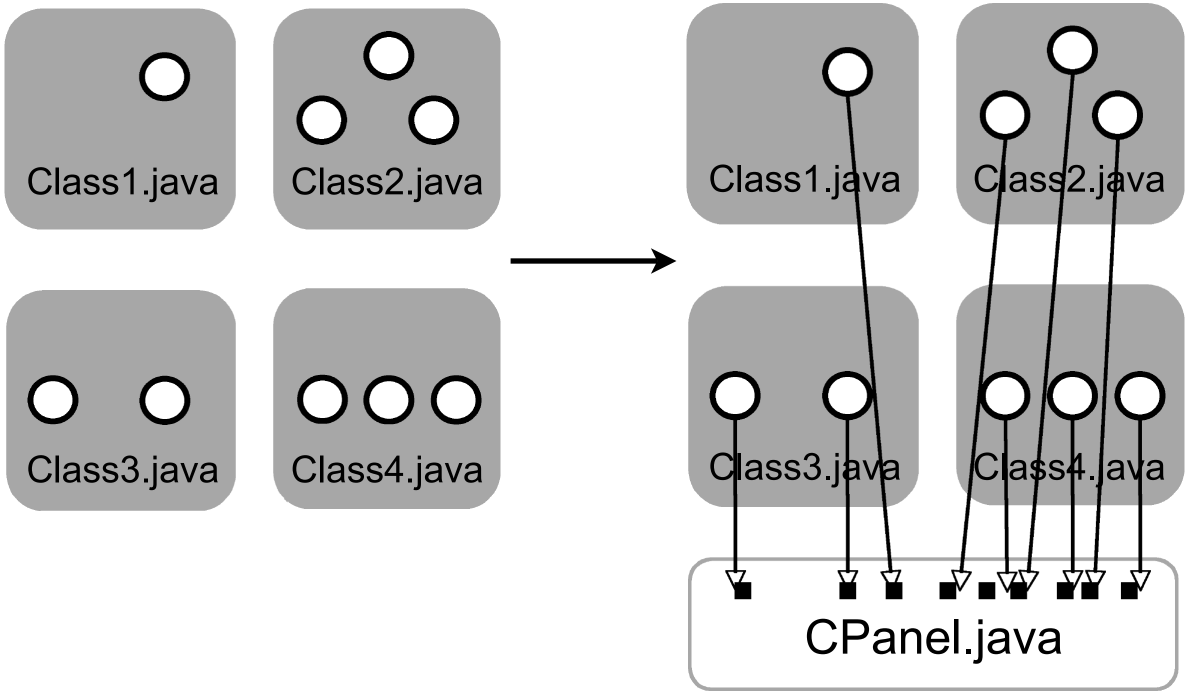
Rearchitecture using TXL (Past)

- Markup declarations with XML
- Normalize source files
- Trace references
- Synthesize control panel



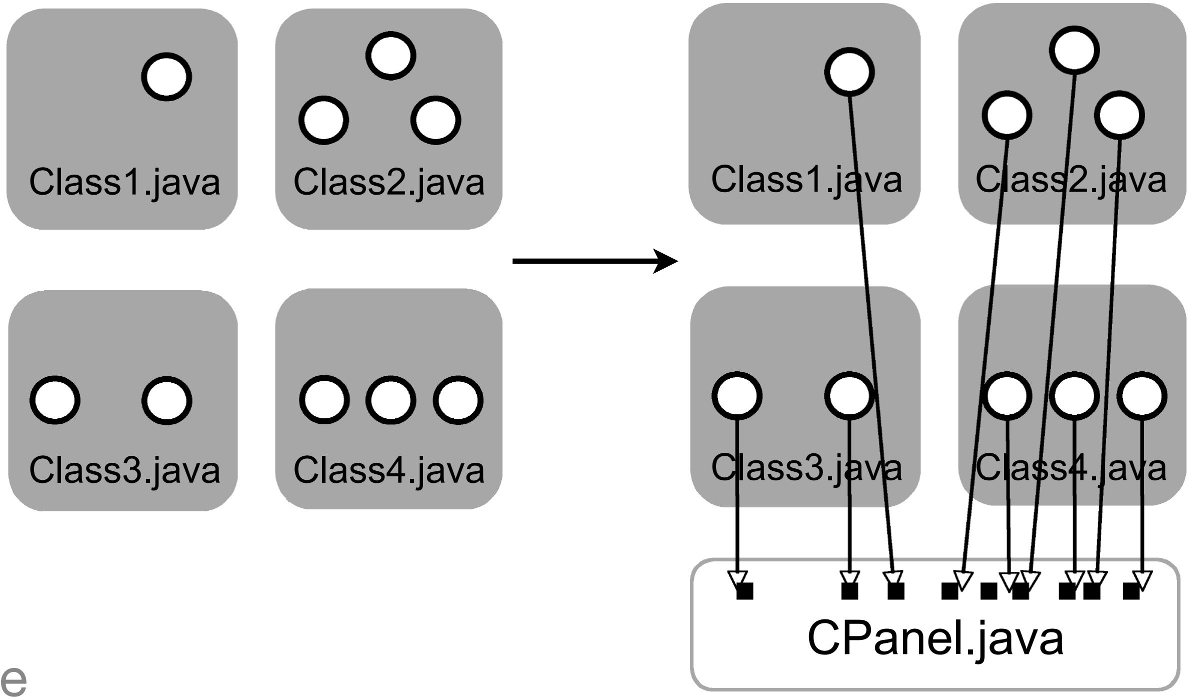
Rearchitecture using TXL (Past)

- Markup declarations with XML
- Normalize source files
- Trace references
- Synthesize control panel
- Redirect references



Rearchitecture using TXL (Past)

- Markup declarations with XML
- Normalize source files
- Trace references
- Synthesize control panel
- Redirect references
- Reconstitute original structure



Example: A Simple Server (Before)

```
public class Server {
    ...
    <control_param>
    private int maxConnections = 100;
    </control_param>
    ...

    public boolean accept() {
        if (numConnections < maxConnections) {
        ...
    }
    ...
}
```

Example: A Simple Server (After)

```
public class Server {
    ...
    CPanel.maxConnections.set(100);
    ...

    public boolean accept() {
        if (numConnections <
            CPanel.maxConnections.get()) {
        ...
    }
    ...
}
```

```
public class CPanel {
    public static class maxConnections {
        private static int value;

        public static int create() {
            return value;
        }

        public static int set(int value) {
            this.value = value;
            return value;
        }

        public static int get() {
            return value;
        }
    }
}
```

Example: A Simple Server

Before	After
<control_param> private int maxConnections = 100; </control_param>	CPanel.maxConnections.set(100);
if (numConnections < maxConnections)	if (numConnections < CPanel.maxConnections.get())

Integration (Future)

Integration (Future)

- Control panel exposes parameters to autonomic controllers

Integration (Future)

- Control panel exposes parameters to autonomic controllers
- Autonomic controllers make decisions and tune parameters

Integration (Future)

- Control panel exposes parameters to autonomic controllers
- Autonomic controllers make decisions and tune parameters
- Generate WSDL specification for control panel to allow management via WSDM

Identification (Present)

Identification (Present)

- Need to automatically identify tuning parameters and markup declarations

Identification (Present)

- Need to automatically identify tuning parameters and markup declarations
- Survey existing server-oriented applications for tuning parameter patterns of use

Identification (Present)

- Need to automatically identify tuning parameters and markup declarations
- Survey existing server-oriented applications for tuning parameter patterns of use
- Iteratively develop and refine a model for pattern analysis

Identification (Present)

- Need to automatically identify tuning parameters and markup declarations
- Survey existing server-oriented applications for tuning parameter patterns of use
- Iteratively develop and refine a model for pattern analysis
- Extract relevant facts and infer new ones (i.e., create a directed graph and manipulate edges using relational algebra)

Identification (Present)

- Need to automatically identify tuning parameters and markup declarations
- Survey existing server-oriented applications for tuning parameter patterns of use
- Iteratively develop and refine a model for pattern analysis
- Extract relevant facts and infer new ones (i.e., create a directed graph and manipulate edges using relational algebra)
- Match patterns (i.e., query graph for instances of patterns)

A Survey of Tuning Parameters

A Survey of Tuning Parameters

- Studied four open-source, server-oriented applications written in Java:
    - Apache Tomcat (Web)
    - Apache Derby (Database)
    - Jetty (Web)
    - Oracle Berkeley DB Java Edition (Database)

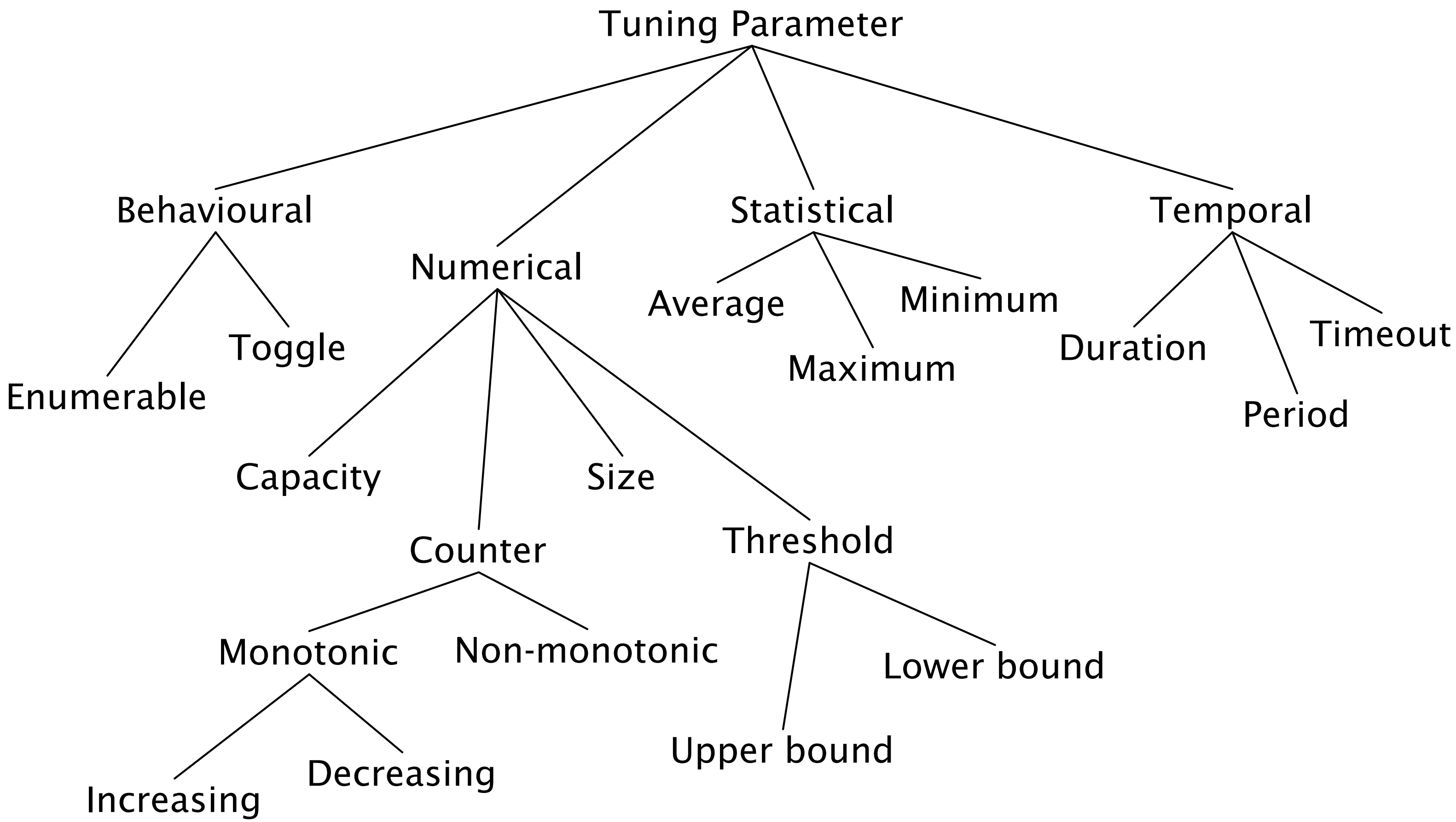
A Survey of Tuning Parameters

- Studied four open-source, server-oriented applications written in Java:
  - Apache Tomcat (Web)
  - Apache Derby (Database)
  - Jetty (Web)
  - Oracle Berkeley DB Java Edition (Database)
- Catalogued the documented tuning parameters from manuals, JavaDoc, and JMX beans

A Survey of Tuning Parameters

- Studied four open-source, server-oriented applications written in Java:
  - Apache Tomcat (Web)
  - Apache Derby (Database)
  - Jetty (Web)
  - Oracle Berkeley DB Java Edition (Database)
- Catalogued the documented tuning parameters from manuals, JavaDoc, and JMX beans
- Traced source code references and classified them according to patterns of use

Patterns of Use: A Taxonomy



Background: Eclipse JDT

Background: Eclipse JDT

- Eclipse provides a Java parser and in-memory program model with its Java Development Tools (JDT)

Background: Eclipse JDT

- Eclipse provides a Java parser and in-memory program model with its Java Development Tools (JDT)
- Parser generates an abstract syntax tree for a compilation unit (e.g., statements, expressions, declarations)

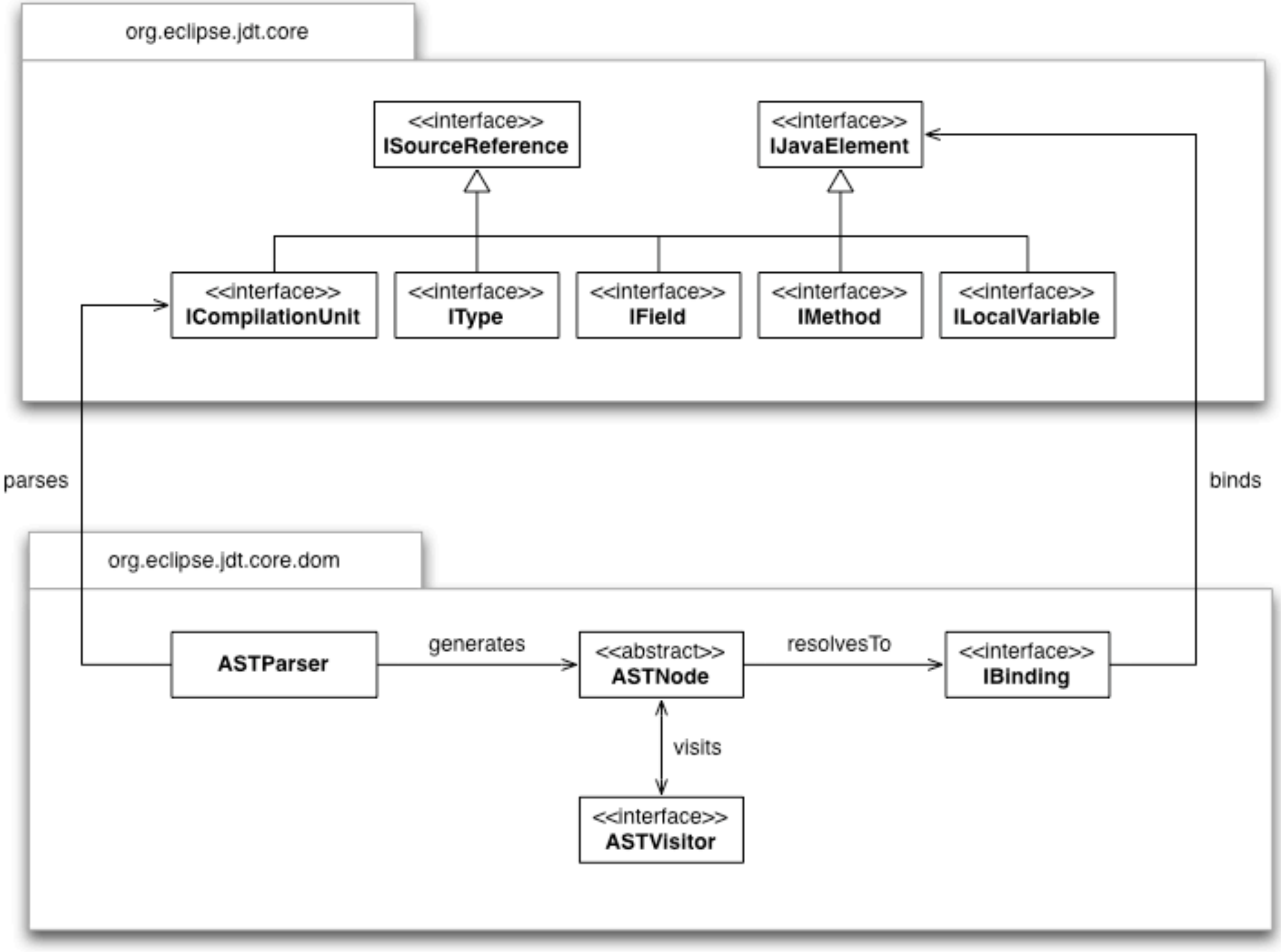
Background: Eclipse JDT

- Eclipse provides a Java parser and in-memory program model with its Java Development Tools (JDT)

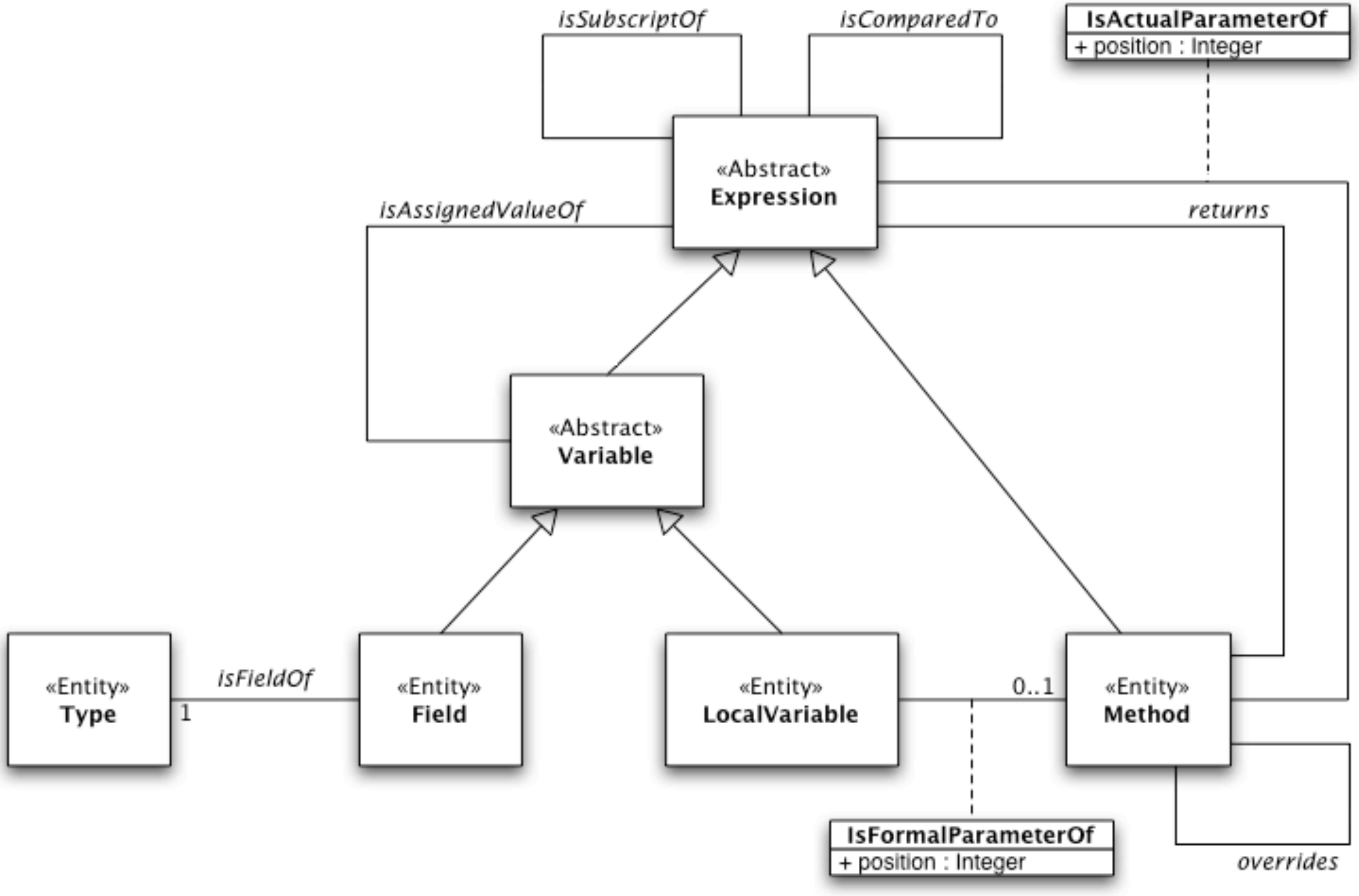
- Parser generates an abstract syntax tree for a compilation unit (e.g., statements, expressions, declarations)

- Program model provides traceability from program model elements to actual source positions (e.g., types, fields, methods, local variables)

Background: Eclipse JDT



An Entity-Relationship Model

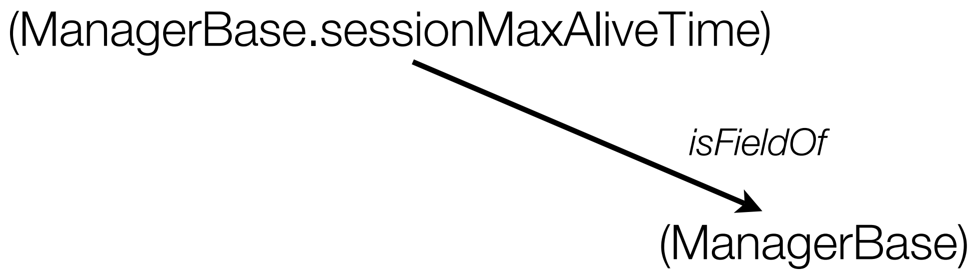


Fact Extraction and Inference

Fact Extraction and Inference

(ManagerBase.sessionMaxAliveTime)

Fact Extraction and Inference

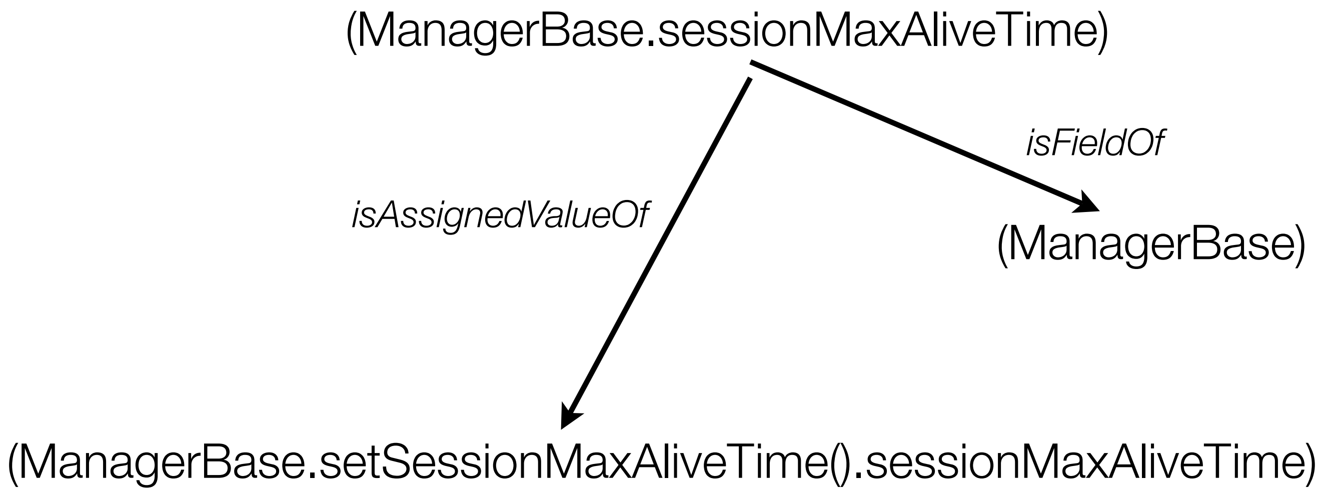


(ManagerBase.sessionMaxAliveTime)

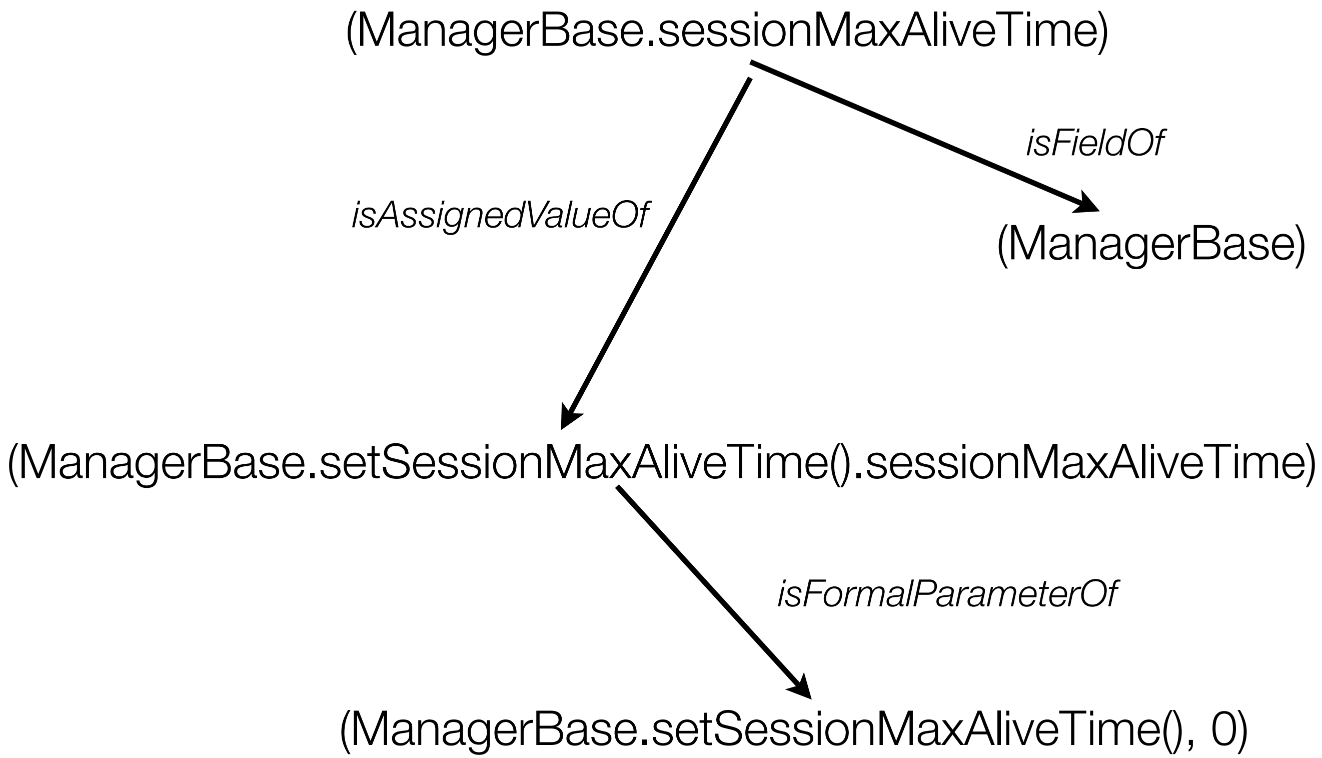
isFieldOf

(ManagerBase)

Fact Extraction and Inference


(ManagerBase.sessionMaxAliveTime)
isFieldOf
isAssignedValueOf
(ManagerBase)
(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

Fact Extraction and Inference



(ManagerBase.sessionMaxAliveTime)

isFieldOf

(ManagerBase)

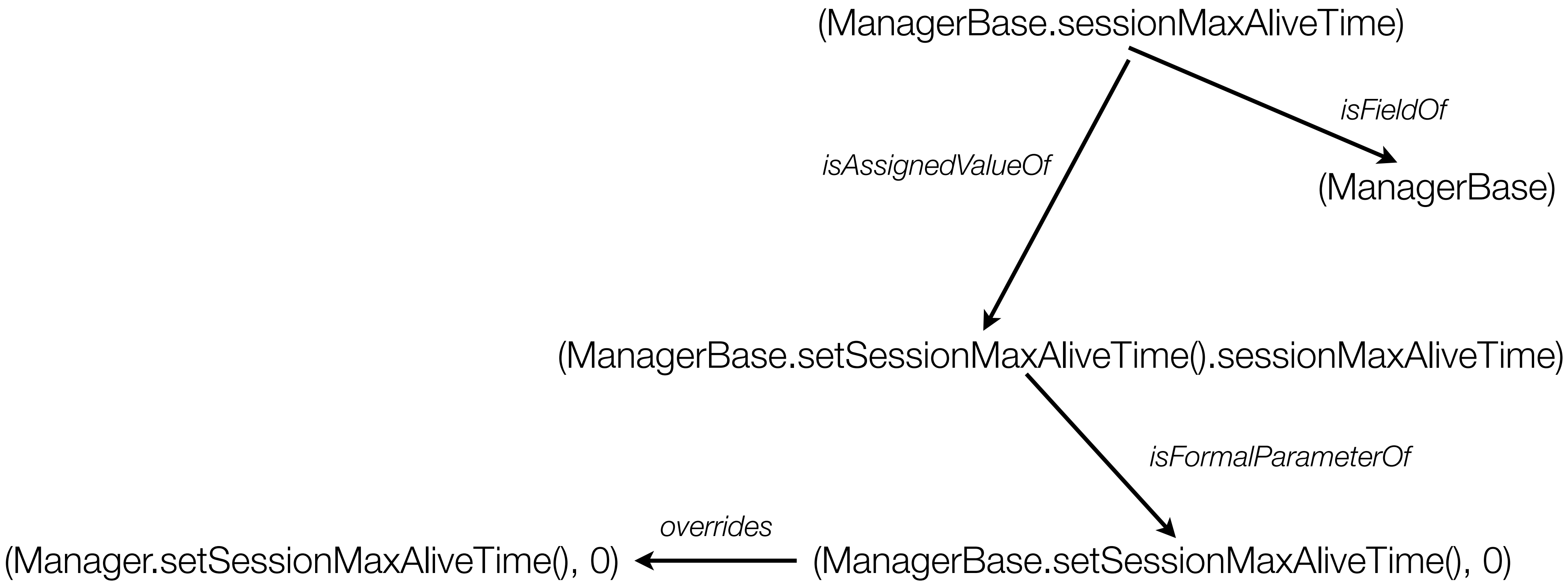
isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isFormalParameterOf

(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(ManagerBase.sessionMaxAliveTime)

isFieldOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

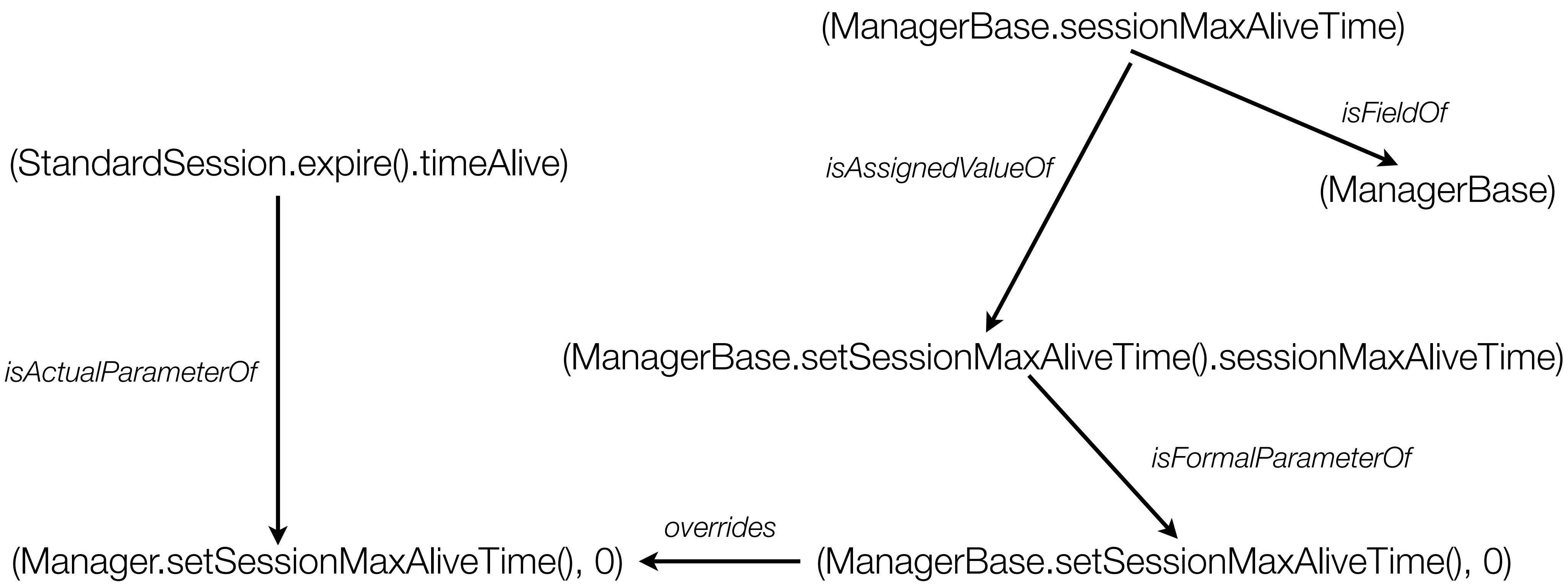
isFormalParameterOf

(ManagerBase.setSessionMaxAliveTime(), 0)

overrides

(Manager.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(ManagerBase.sessionMaxAliveTime)

isFieldOf

(ManagerBase)

isAssignedValueOf

(StandardSession.expire().timeAlive)

isActualParameterOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

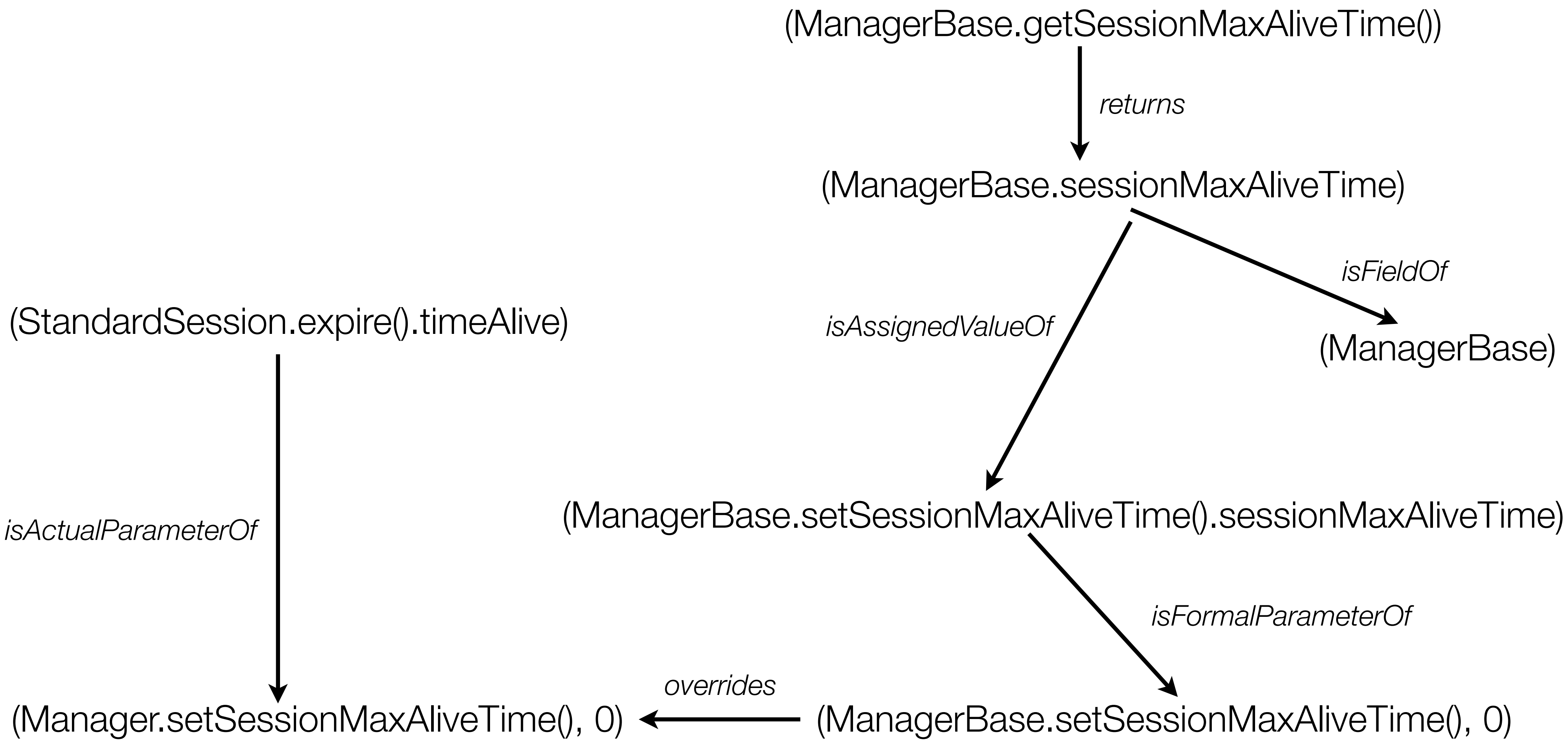
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

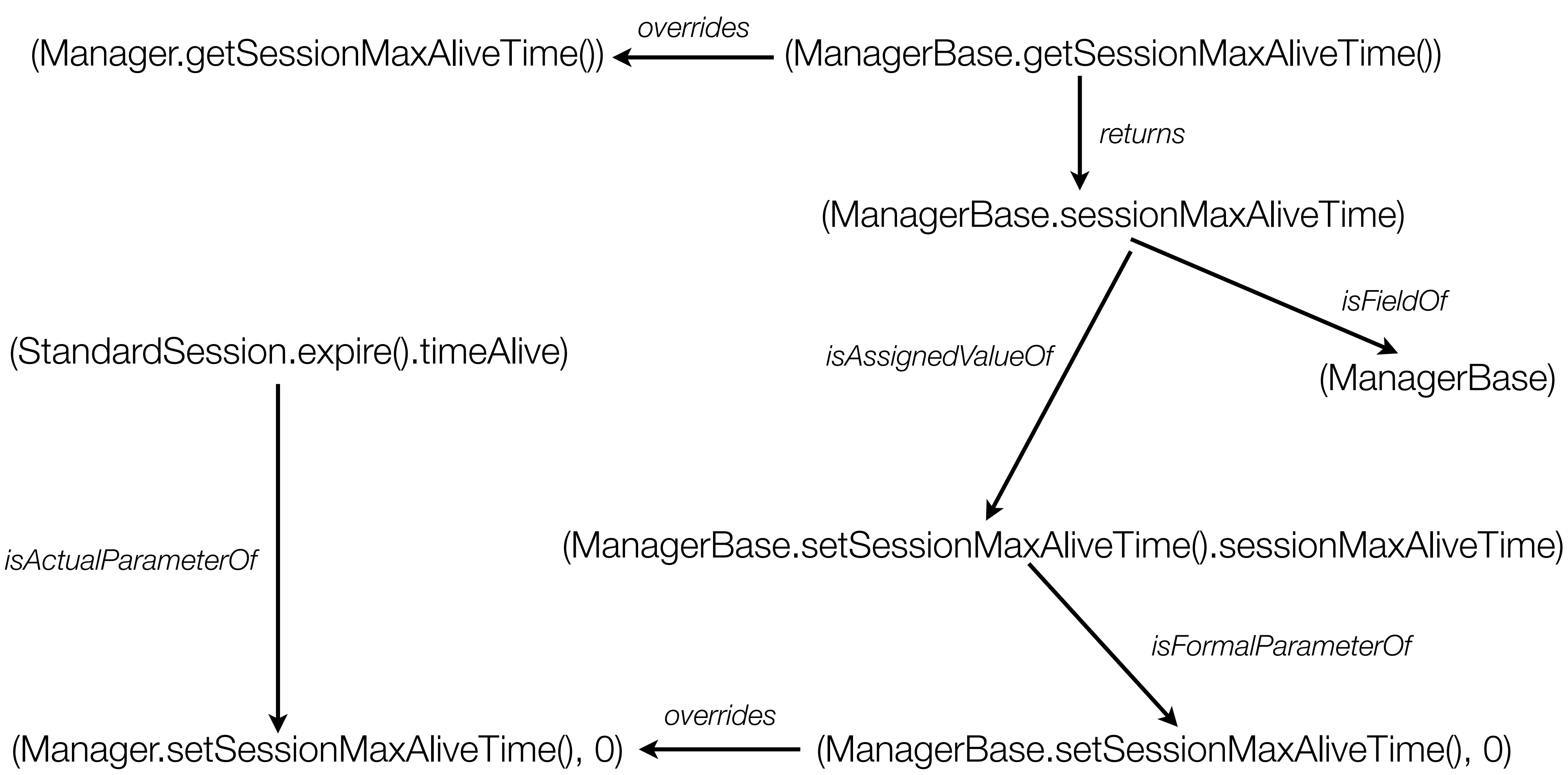
(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



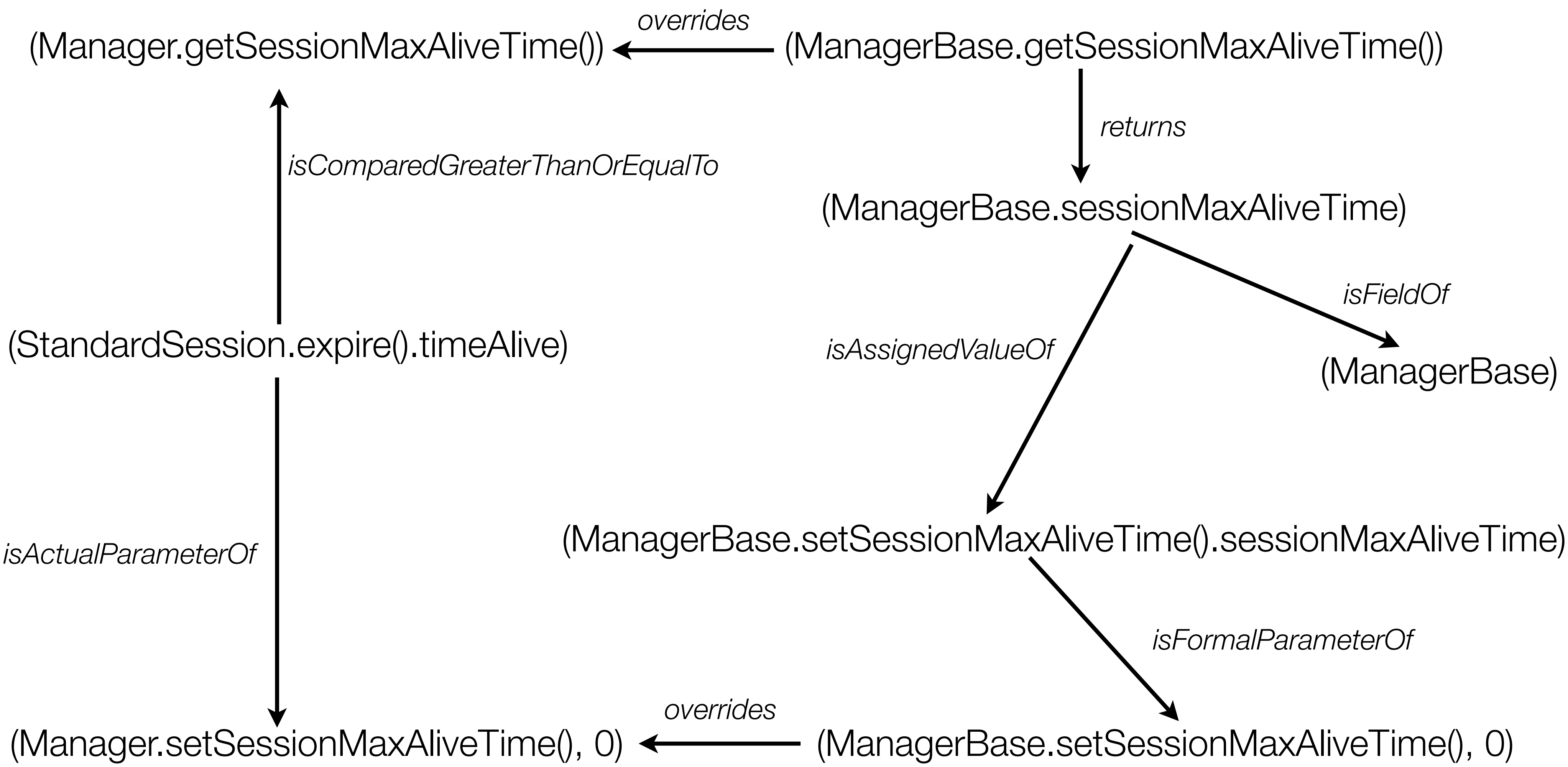
(ManagerBase.getSessionMaxAliveTime())
returns
(ManagerBase.sessionMaxAliveTime)
isFieldOf
(ManagerBase)
isAssignedValueOf
(StandardSession.expire().timeAlive)
isActualParameterOf
(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)
isFormalParameterOf
overrides
(Manager.setSessionMaxAliveTime(), 0)
(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())
overrides
(ManagerBase.getSessionMaxAliveTime())
returns
(ManagerBase.sessionMaxAliveTime)
isFieldOf
(ManagerBase)
isAssignedValueOf
(StandardSession.expire().timeAlive)
isActualParameterOf
(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)
isFormalParameterOf
overrides
(Manager.setSessionMaxAliveTime(), 0)
(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isFieldOf

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

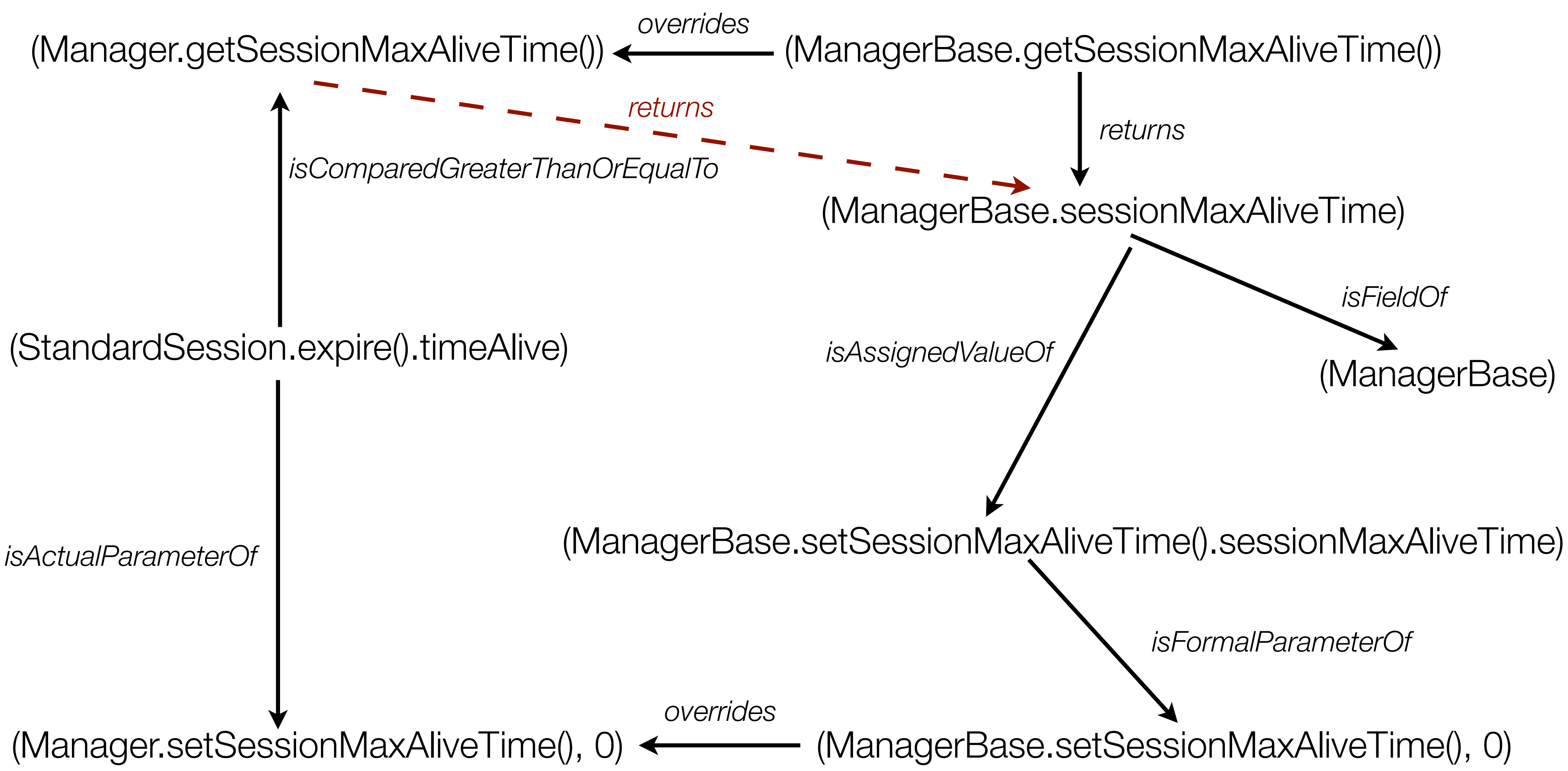
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isFieldOf

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

isActualParameterOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

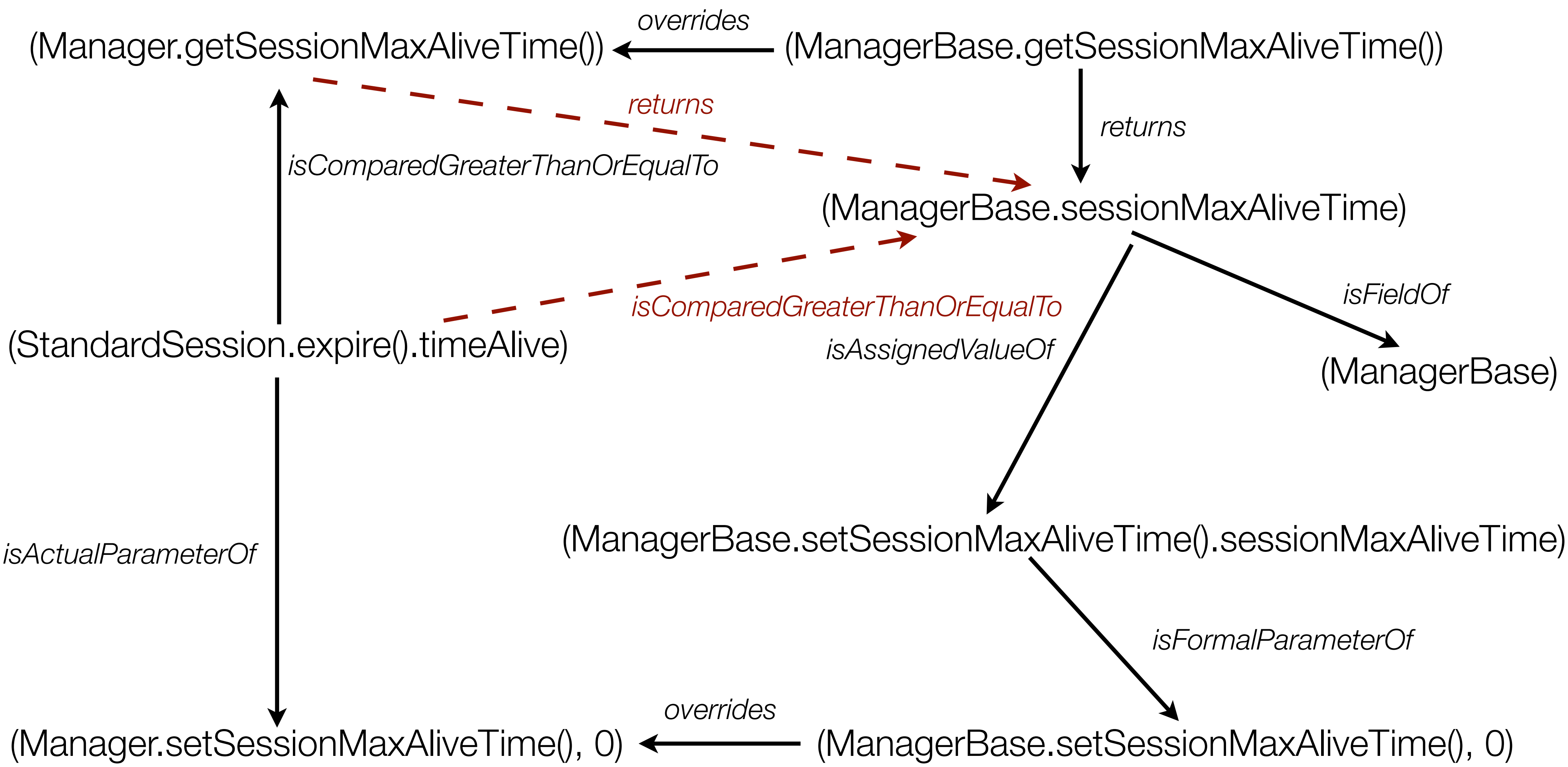
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isFieldOf

(StandardSession.expire().timeAlive)

isComparedGreaterThanOrEqualTo

isAssignedValueOf

(ManagerBase)

isActualParameterOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

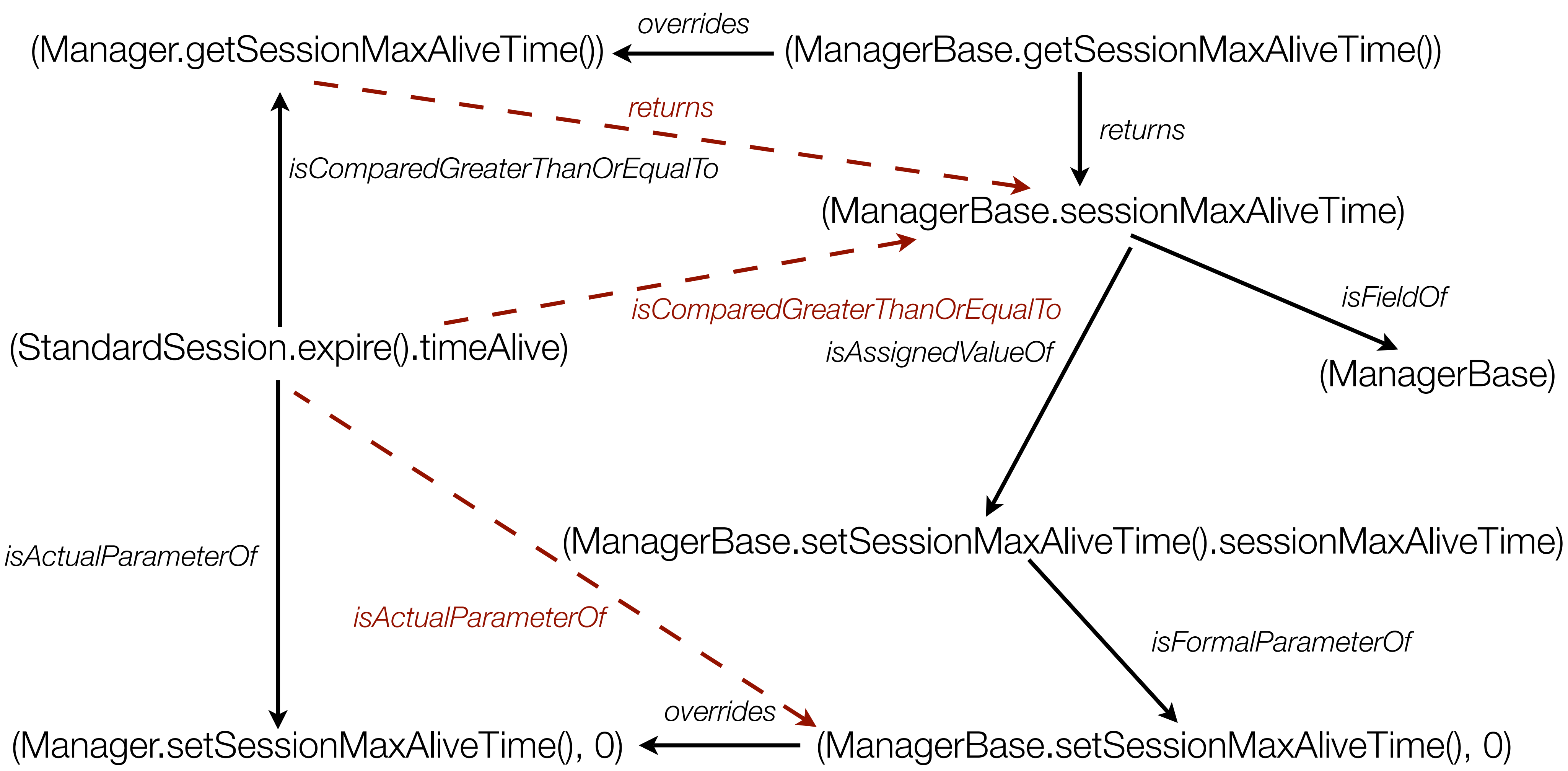
isFormalParameterOf

overrides

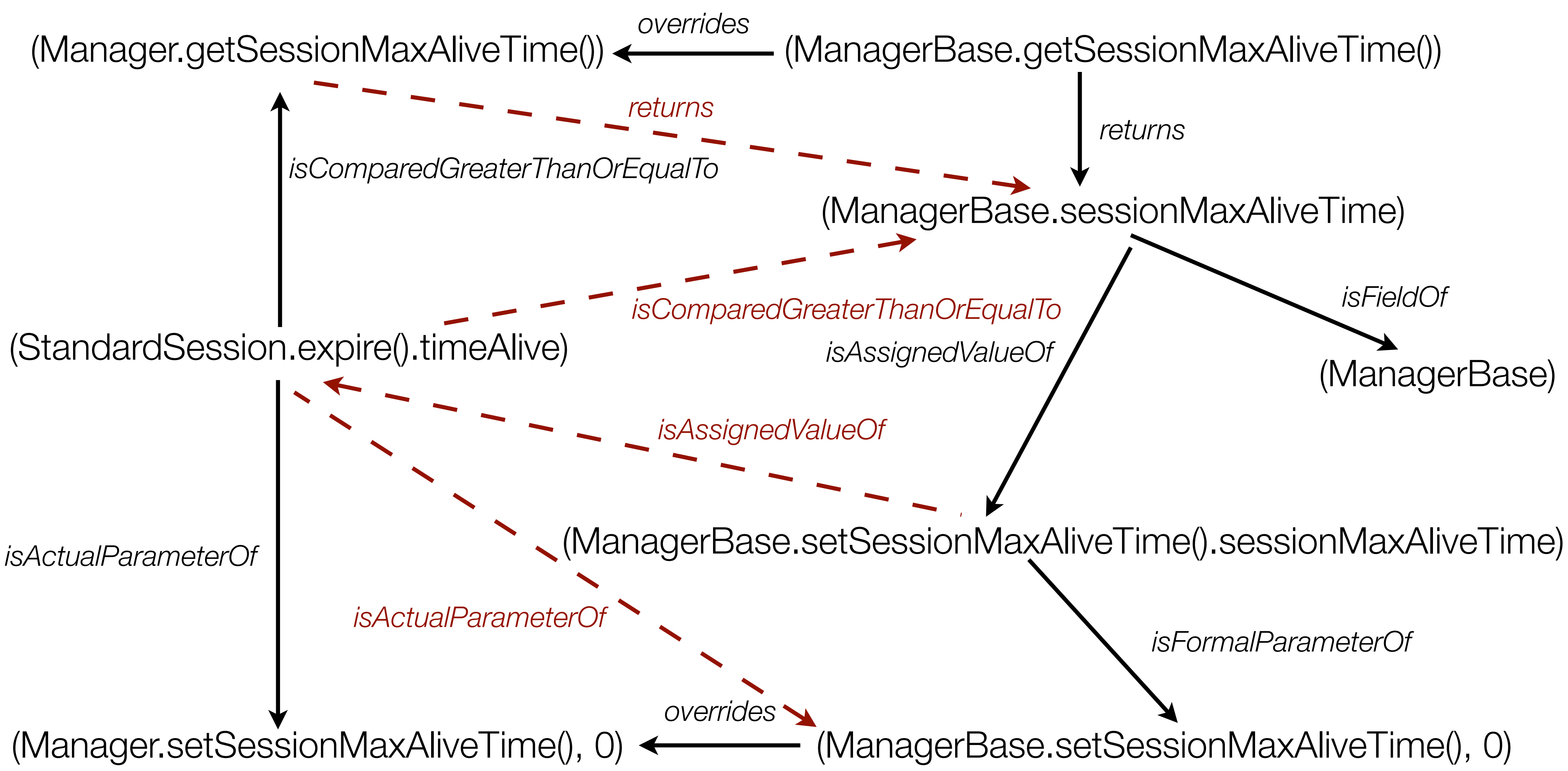
(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference


(Manager.getSessionMaxAliveTime())
overrides
(ManagerBase.getSessionMaxAliveTime())
returns
returns
isComparedGreaterThanOrEqualTo
(ManagerBase.sessionMaxAliveTime)
isFieldOf
(StandardSession.expire().timeAlive)
isComparedGreaterThanOrEqualTo
isAssignedValueOf
(ManagerBase)
isActualParameterOf
(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)
isActualParameterOf
isFormalParameterOf
overrides
(Manager.setSessionMaxAliveTime(), 0)
(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isFieldOf

isComparedGreaterThanOrEqualTo

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

isActualParameterOf

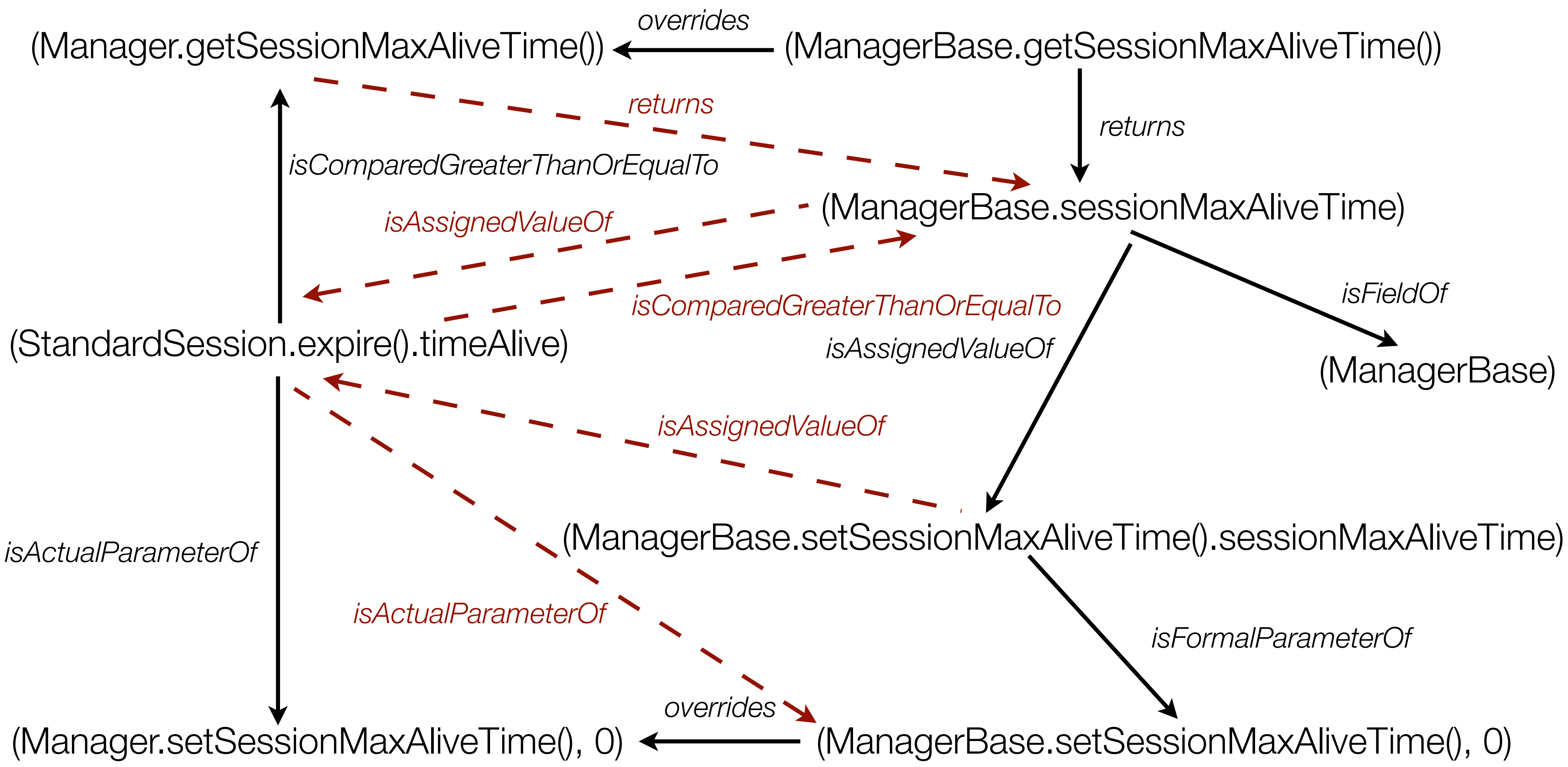
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Fact Extraction and Inference



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isAssignedValueOf

isFieldOf

(StandardSession.expire().timeAlive)

isComparedGreaterThanOrEqualTo

isAssignedValueOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

isActualParameterOf

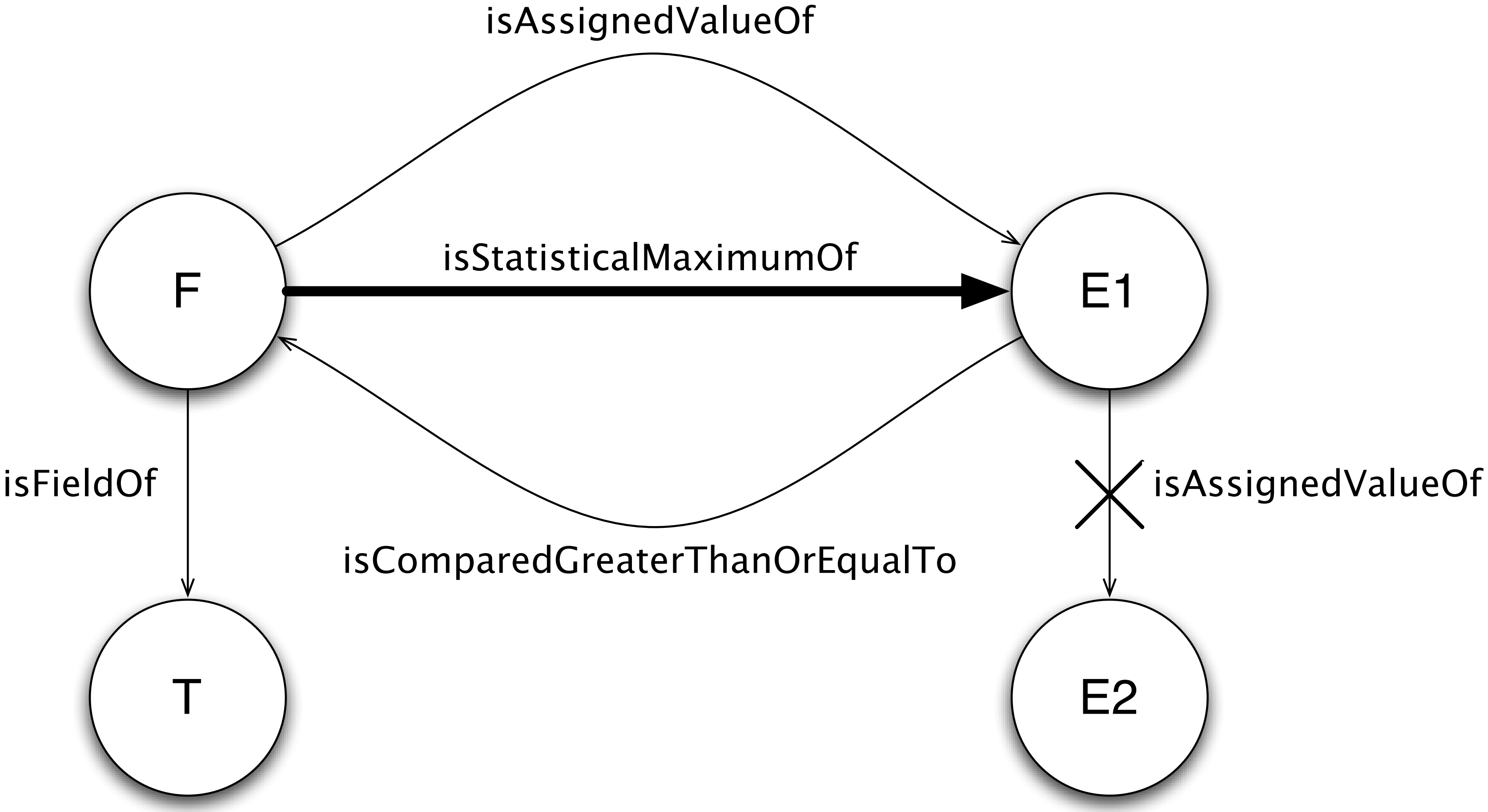
isFormalParameterOf

overrides

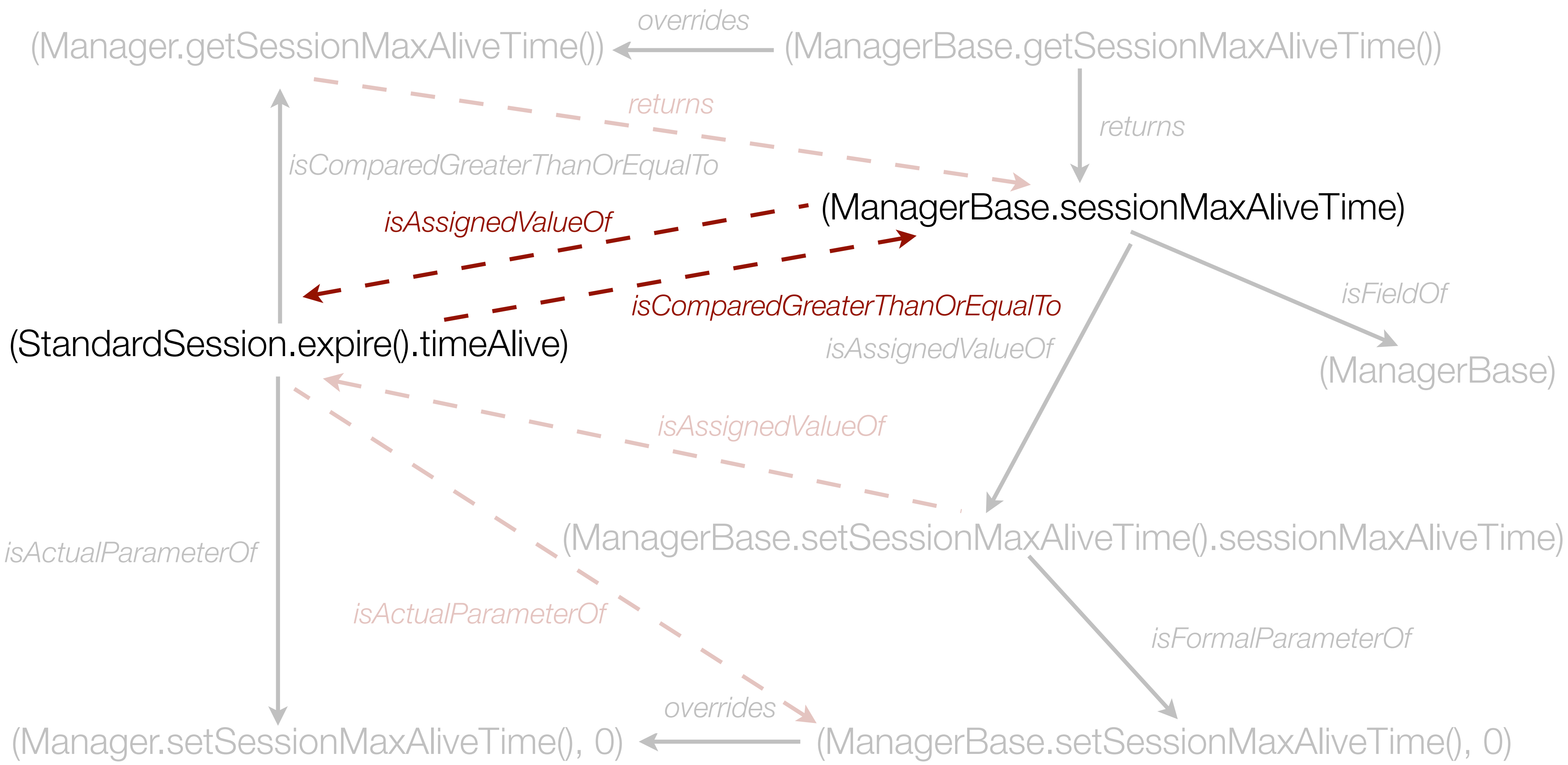
(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Graph Pattern Matching: Statistical Maximum



Graph Pattern Matching



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

isAssignedValueOf

(ManagerBase.sessionMaxAliveTime)

isFieldOf

isComparedGreaterThanOrEqualTo

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

isActualParameterOf

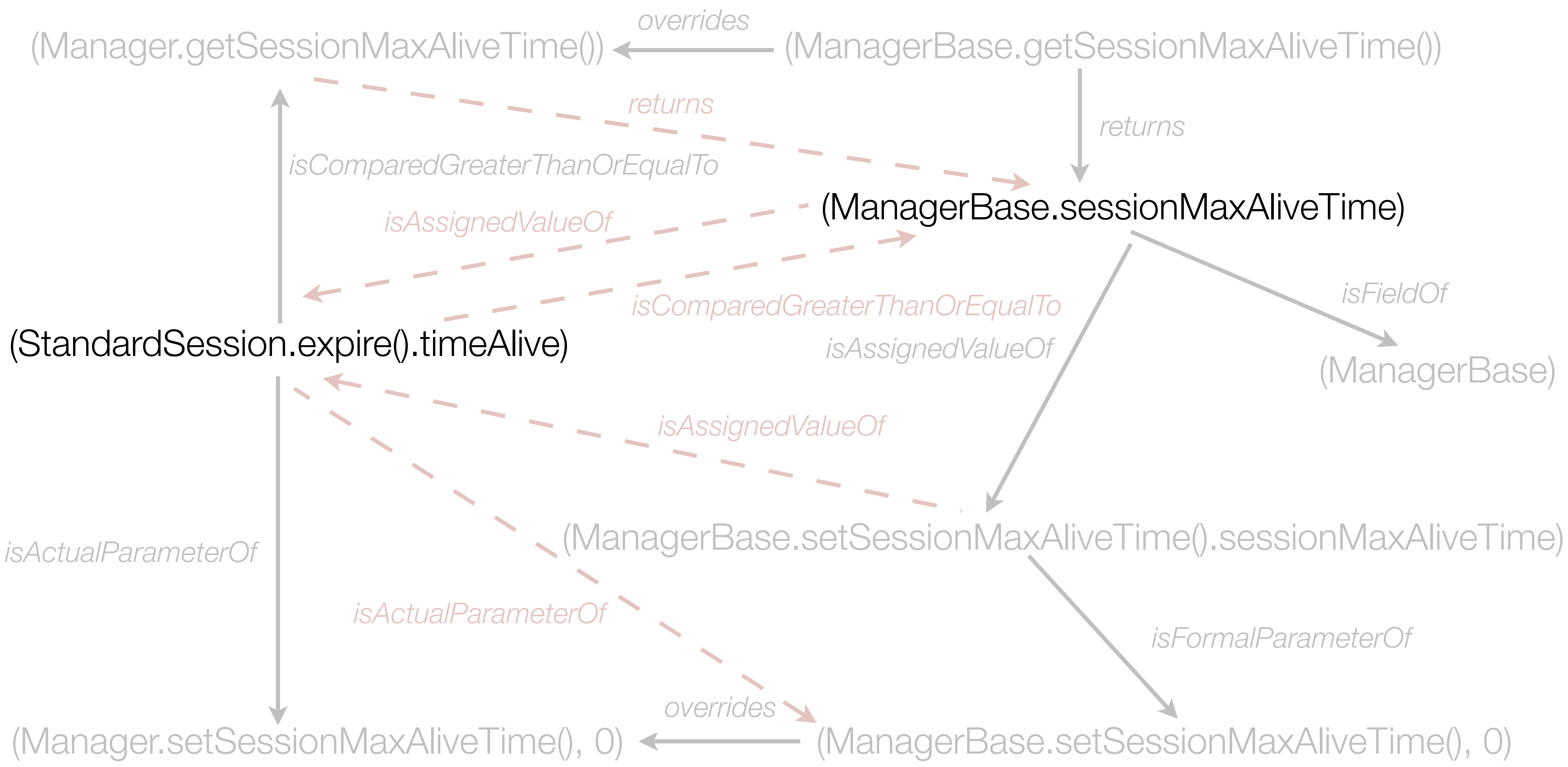
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Graph Pattern Matching



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isAssignedValueOf

isFieldOf

isComparedGreaterThanOrEqualTo

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

isActualParameterOf

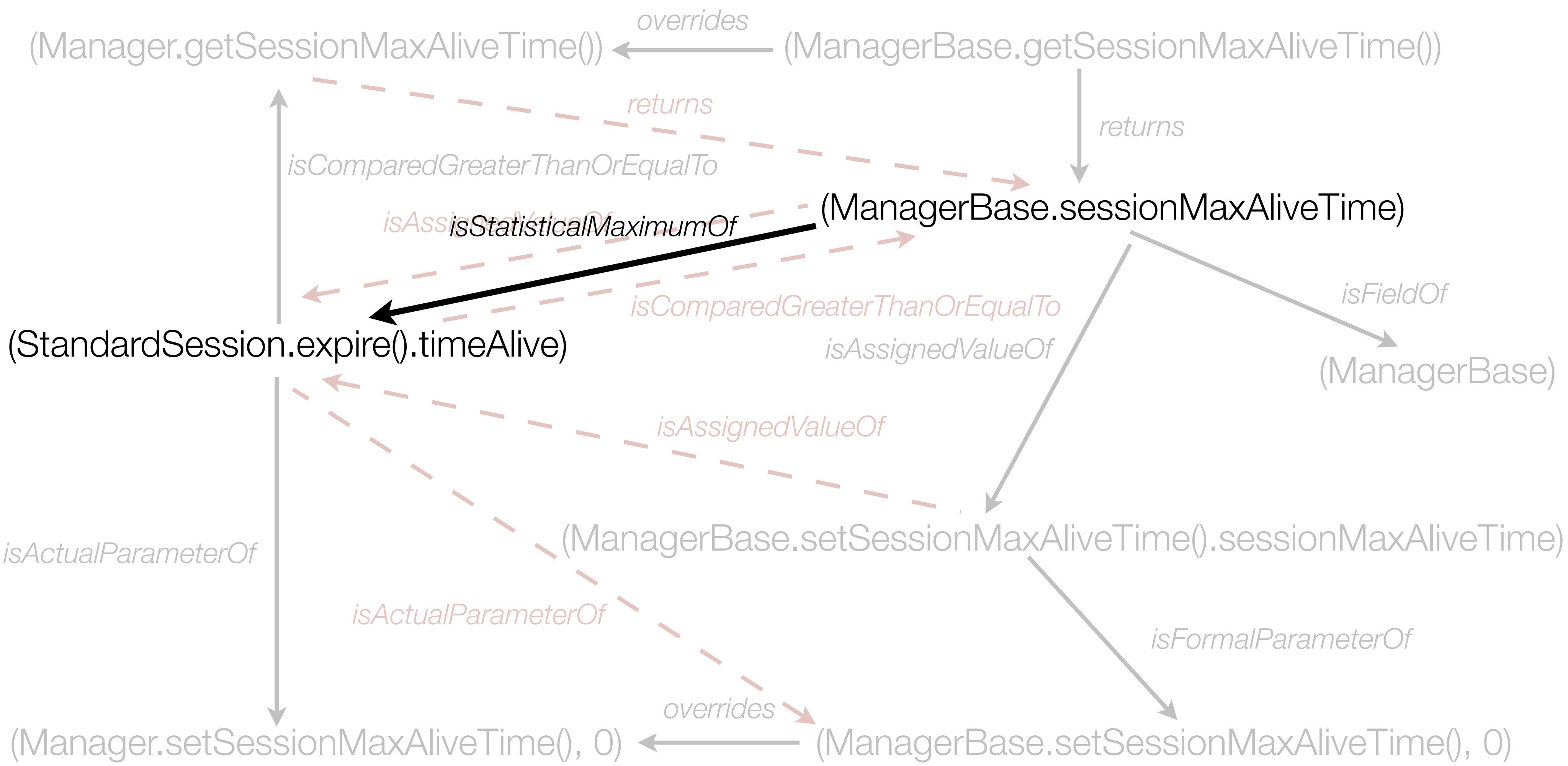
isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Graph Pattern Matching



(Manager.getSessionMaxAliveTime())

overrides

(ManagerBase.getSessionMaxAliveTime())

returns

returns

isComparedGreaterThanOrEqualTo

(ManagerBase.sessionMaxAliveTime)

isAssignedValueOf

isStatisticalMaximumOf

isComparedGreaterThanOrEqualTo

isFieldOf

(StandardSession.expire().timeAlive)

isAssignedValueOf

(ManagerBase)

isAssignedValueOf

(ManagerBase.setSessionMaxAliveTime().sessionMaxAliveTime)

isActualParameterOf

isActualParameterOf

isFormalParameterOf

overrides

(Manager.setSessionMaxAliveTime(), 0)

(ManagerBase.setSessionMaxAliveTime(), 0)

Results

Results

- Implemented an Eclipse plug-in that extends the search framework to find tuning parameters: counter, size, statistical maximum

Results

- Implemented an Eclipse plug-in that extends the search framework to find tuning parameters: counter, size, statistical maximum

- Identifies catalogued tuning parameters along with some undocumented ones

Results

- Implemented an Eclipse plug-in that extends the search framework to find tuning parameters: counter, size, statistical maximum

- Identifies catalogued tuning parameters along with some undocumented ones

- Successfully applied to another application domain (i.e., instant messaging server)

Future Work

Future Work

- Formal evaluation using applications not included in survey with more comprehensive set of tuning parameter patterns of use

Future Work

- Formal evaluation using applications not included in survey with more comprehensive set of tuning parameter patterns of use

- Taxonomy needs to be refined based on expert feedback, other application domains and patterns of use for implicit tuning patterns

Future Work

- Formal evaluation using applications not included in survey with more comprehensive set of tuning parameter patterns of use

- Taxonomy needs to be refined based on expert feedback, other application domains and patterns of use for implicit tuning patterns

- Source transformations to make implicit tuning parameters explicit (e.g., tree depths, cache hit rates)

Future Work

- Formal evaluation using applications not included in survey with more comprehensive set of tuning parameter patterns of use

- Taxonomy needs to be refined based on expert feedback, other application domains and patterns of use for implicit tuning patterns

- Source transformations to make implicit tuning parameters explicit (e.g., tree depths, cache hit rates)

- Orthogonal pattern characterisations based on resource stereotypes

References

[1] Elizabeth Dancy, James R. Cordy, “STAC: Software Tuning Panels for Autonomic Control,” CASCON, pp. 146-160, 2006 Conference of the Center For Advanced Studies on Collaborative Research, 2006.

[2] Jeffrey O. Kephart, David M. Chess, "The Vision of Autonomic Computing," Computer, vol. 36, no. 1, pp. 41-50, Jan., 2003.

[3] Andrew Malton, Kevin A. Schneider, James R. Cordy, Thomas. R. Dean, Darren Cousineau, Jason Reynolds, "Processing Software Source Text in Automated Design Recovery and Transformation," IWPC, p. 127, Ninth International Workshop on Program Comprehension, 2001.

[4] Thomas R. Dean, James R. Cordy, Kevin A Schneider, Andrew J. Malton, "Using Design Recovery Techniques to Transform Legacy Systems," ICSM, p. 622, 17th IEEE International Conference on Software Maintenance, 2001.

[5] R.C. Holt, "Binary Relational Algebra Applied to Software Architecture," Technical Report 345, Computer Science Research Institute, University of Toronto, March 1996.

[6] M.P. Consens and A.O. Mendelzon, "GraphLog: A Visual Formalism for Real Life Recursion," PODS '90, Nashville, April 1990, pp. 404–416.

[7] IBM Research Autonomic Computing Overview. Retrieved March 12, 2007 from http://www.research.ibm.com/autonomic/overview/faqs.html#8.