



Prediction Enabled Component Technology 1

Component
Technology

We begin with vanilla software
component technology

- marked adoption trend
- .NET, J2EE, custom exemplars

Motivation for adoption include:

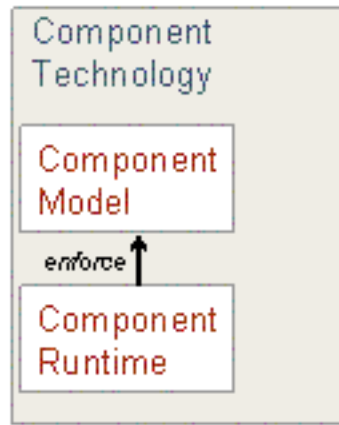
- larger, reusable building blocks
- reusable design patterns, rules
- simplified system integration

[Next](#)

[Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)

Prediction Enabled Component Technology 2



Component model

- required component interfaces and other implementation constraints
- behavior specifications
 - component type-specific, general
 - lifecycle and resource management policies

Component runtime environment

- runtime and deployment services
- execution environment ("container")
- runtime policy enforcement

There are many views on component technology

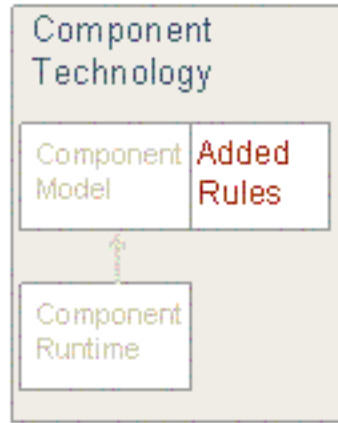
- our view is representative of theory and practice

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 3



We extend the base component model

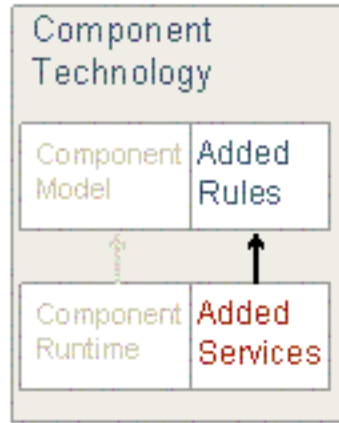
- these impose additional rules on component/assembly developer

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 4



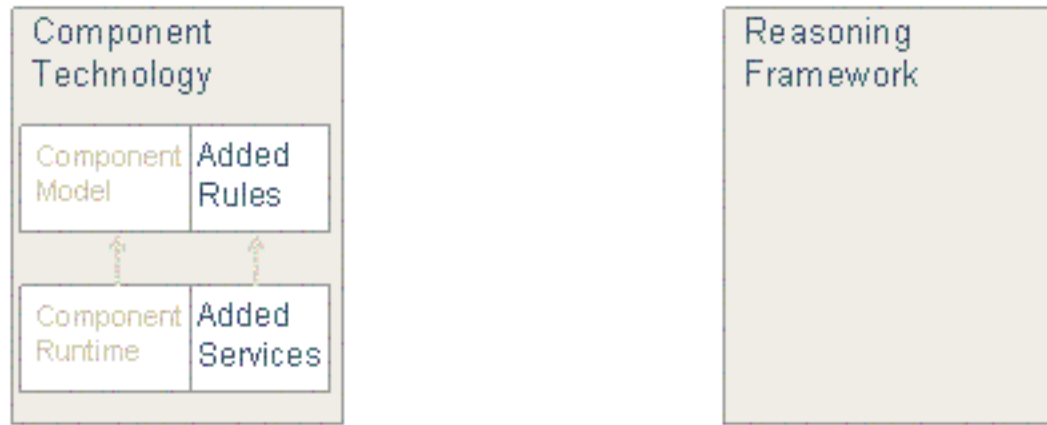
Where useful, and possible, we augment the runtime as well

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 5



The added rules and services are not arbitrary, but are imposed by a reasoning framework

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)

Prediction Enabled Component Technology 6

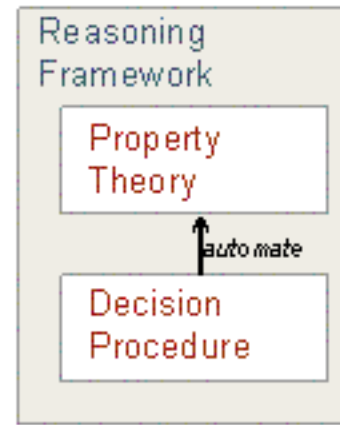
Reasoning frameworks provide automated prediction services

Property theory

- logic and notation for reasoning about assembly properties

Decision procedure

- an automated procedure for computing predicted properties



Reasoning frameworks can be defined for any objectively observable runtime assembly property

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)

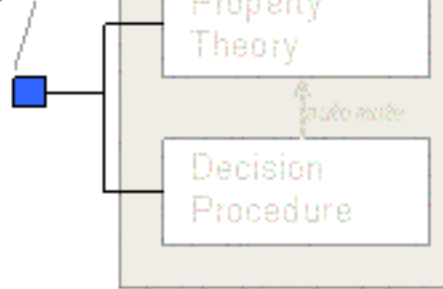
Prediction Enabled Component Technology 7

A reasoning framework will have assumptions about the systems it models

- scheduling
- use conditions
- physical environment
- etc.

Our approach makes these assumptions explicit

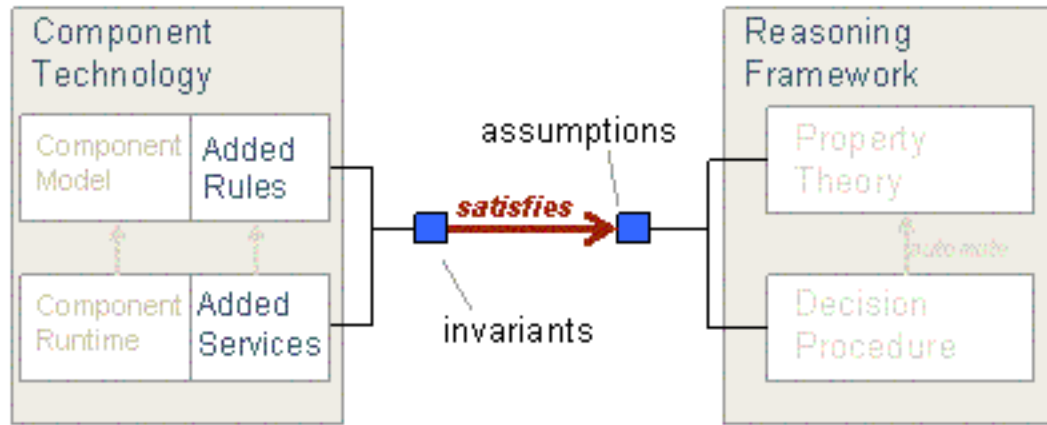
assumptions



[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)

Prediction Enabled Component Technology 8



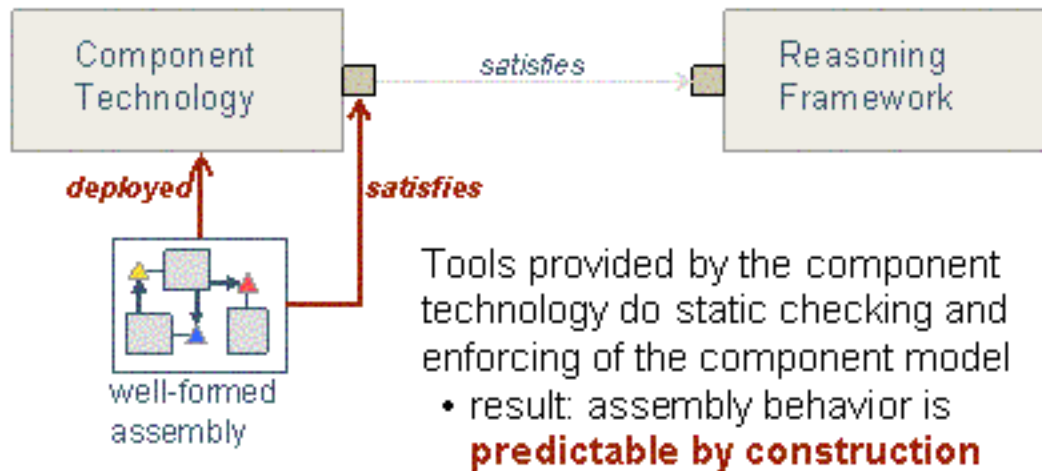
The component technology ensures that components & assemblies satisfy reasoning framework assumptions

[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 9

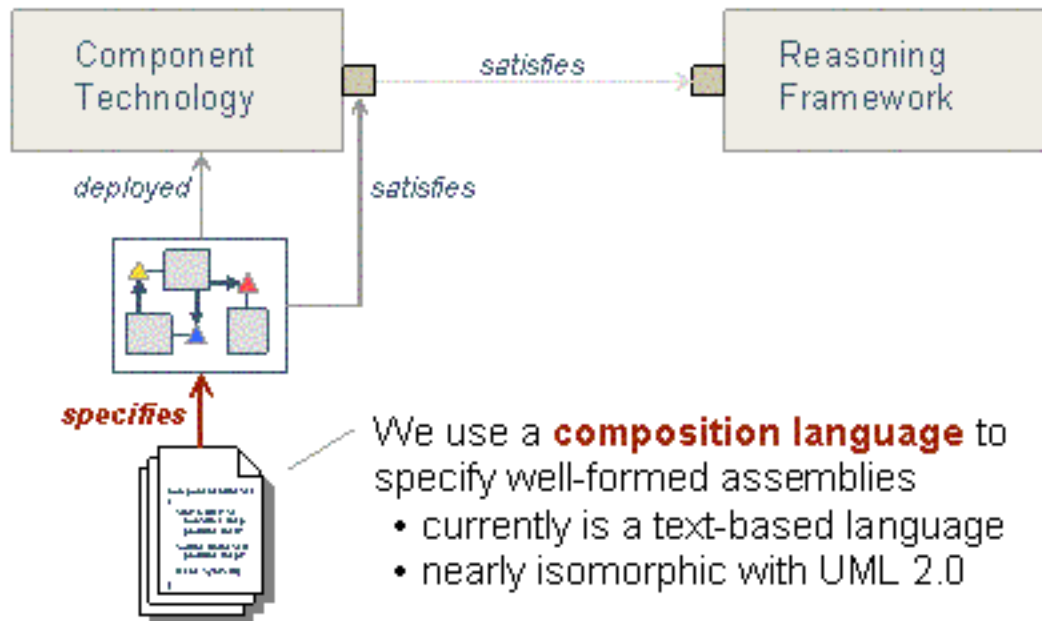


[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 10

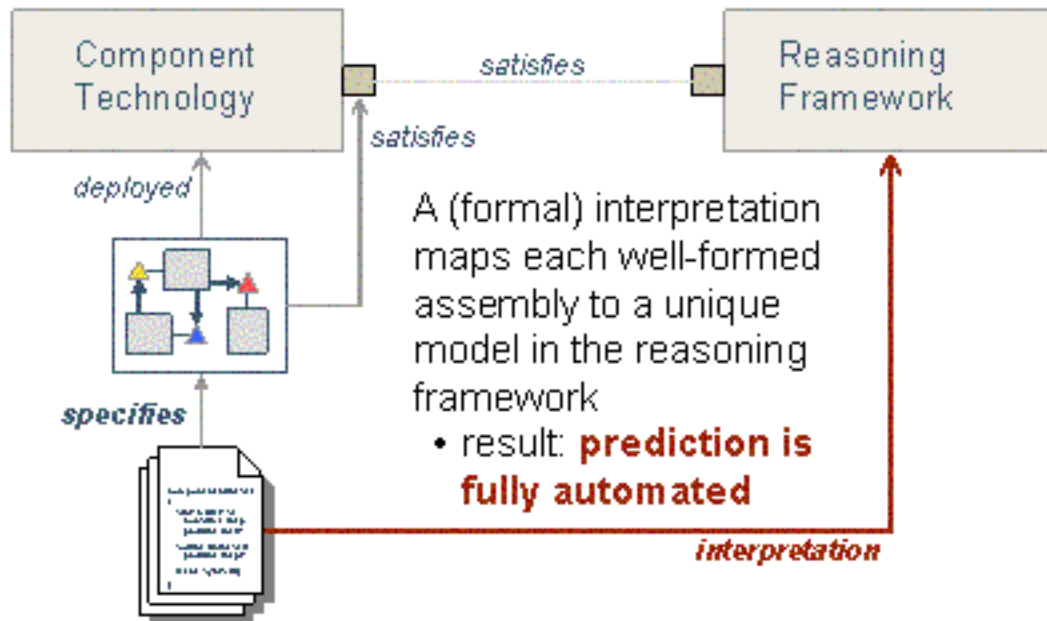


[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 11



[Previous](#) [Next](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)



Prediction Enabled Component Technology 12



A PECT can have N reasoning frameworks, each of which

- will predict some observable assembly runtime property
- will have its own interpretation and decision procedure
- may introduce its own checkable well-formedness constraints on components and their assemblies

[Previous](#) [Exit Slide Show](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#)