

Globale Analyse: Entwickle Strategien I

1. Identifiziere Probleme und deren Einflussfaktoren

Analysiere Faktorentabelle:

- Grenzen oder Einschränkungen durch Faktoren
 - *Unverrückbarer Abgabetermin erlaubt keinen vollen Funktionsumfang*
- Notwendigkeit, Auswirkung eines Faktors zu begrenzen
 - *Entwurf muss Portierbarkeit vorsehen*
- Schwierigkeit, einen Produktfaktor zu erfüllen
 - *Speicher- und Prozessorbegrenzung erlaubt keine beliebig komplexen Algorithmen*
- Notwendigkeit einer allgemeinen Lösung zu globalen Anforderungen wie Fehlerbehandlung und Wiederaufsetzen nach Fehlern

2. Entwickle Lösungen und spezifische Strategien

... für die Behandlung der Probleme, die sich implementieren lassen und die notwendige Änderbarkeit unterstützen.

Strategie muss konsistent sein zu:

- Einflussfaktor,
- dessen Veränderlichkeit
- und dessen Interaktion mit anderen Faktoren.

2. Entwickle Lösungen und spezifische Strategien

Ziele:

- Reduzierung oder Kapselung des Faktoreinflusses
- Reduzierung der Auswirkung einer Änderung des Faktors auf den Entwurf und andere Faktoren
- Reduzierung oder Kapselung notwendiger Bereiche von Expertenwissen oder -fähigkeiten
- Reduzierung der Gesamtentwicklungsdauer und -kosten

3. Identifiziere verwandte Strategien

Globale Analyse: Entwickle Strategien IV

Problemkarten (Issue Cards) beschreiben Problem und passende Strategien einmalig:

Name des Problems
<i>Beschreibung des Problems</i> Einflussfaktoren <i>Liste aller Einflussfaktoren</i>
Lösung <i>Diskussion einer allgemeinen Lösung</i> Strategie: Name der Strategie A <i>Erläuterung der Strategie A</i> Strategie: Name der Strategie B <i>Erläuterung der Strategie B</i> ...
Verwandte Strategien <i>Referenzen zu verwandten Strategien.</i> <i>Diskussion, in welcher Weise sie verwandt sind.</i>

Ambitionierter Zeitplan

Abgabetermin ist fix, Ressourcen sind begrenzt. Möglicherweise können nicht alle Produktfunktionen realisiert werden.

Einflussfaktoren

O1.1 : Time-To-Market

O1.2 : Auslieferung von Produktfunktionen

O2.1 : Anzahl Entwickler

Lösung

Strategie: Schrittweiser Ausbau

System wird schrittweise ausgebaut. Anforderungen werden in der Reihenfolge ihrer Priorität realisiert.

Strategie: Einkaufen statt Selbstentwickeln

Einbindung externer COTS-Komponenten.

Strategie: Wiederverwendung

Einbindung interner wiederverwendbarer Komponenten.

Änderungen in allgemeiner und domänenspezifischer Hardware

Regelmäßige Änderungen der Hardware; Software soll mit minimalen Aufwand an Änderungen angepasst werden können.

Einflussfaktoren

T1.1: Prozessoren werden leistungsfähiger.

T2.1: Aufzeichnungs-Hardware ändert sich alle drei Jahre

T2.2: Netzwerktechnologie ändert sich alle vier Jahre

Lösung

Strategie: Kapselung domänenspezifischer Hardware:

Kapsle Details, die sich ändern können.

Strategie: Kapselung allgemeiner Hardware:

Kapsle Details, die sich ändern können.

Betriebssystemanpassung

Weiterentwicklung des Betriebssystems (BS) und Wechsel auf ein neues BS machen Anpassungen an betriebssystemspezifischen Komponenten notwendig. Für die Wartung steht nur ein Entwickler zur Verfügung. Die Anpassungen müssen schnell vorgenommen werden.

Einflussfaktoren

O1.1: Time-To-Market

O2.1: Anzahl Entwickler

T2.3: Betriebssystem

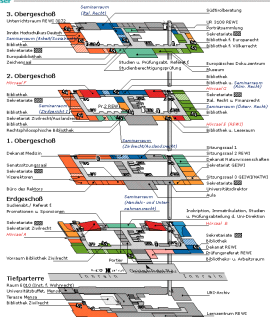
Lösung

Strategie: Kapselung der BS-abhängigen Anteile

BS-abhängige Anteile werden in Komponenten isoliert. Diese bilden eine virtuelle Maschine für alle anderen Komponenten.



Gebäudeplan: Universitätshauptgebäude (Bauteil VIII),
Uni Innsbruck, A-6020 Innsbruck, Innrain 52



Farblegende siehe nächste Seite.
Zugang Lift: Ein führt Tiefgarage, danach links, Innenhof



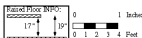
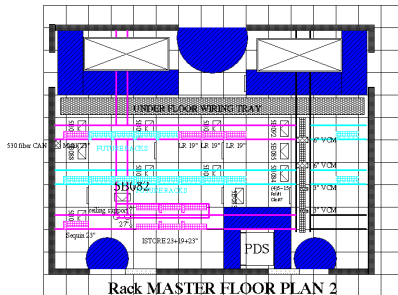
1. Obergeschoss



Erdschoß



Symbollegende siehe unter www.wegweiser.ac.at/symbollegende.html



Scale factor = 4.0 1" = 4'	Latex: 11.11.99 By Phil CUSG
-------------------------------	---------------------------------

530/N530

Definition

Architektursicht (View): Repräsentation eines ganzen Systems aus der Perspektive einer kohärenten Menge von Anliegen (IEEE P1471 2002).

Architektursichten und -blickwinkel

Definition

Architektursicht (View): Repräsentation eines ganzen Systems aus der Perspektive einer kohärenten Menge von Anliegen (IEEE P1471 2002).

Definition

Architekturblickwinkel (Viewpoint): Spezifikation der Regeln und Konventionen, um eine Architektursicht zu konstruieren und zu benutzen (IEEE P1471 2002).

Ein Blickwinkel ist ein Muster oder eine Vorlage, von der aus individuelle Sichten entwickelt werden können, durch Festlegung von

- Zweck,
- adressierte Betrachter,
- und Techniken für Erstellung, Gebrauch und Analyse.

Architektursichten und -blickwinkel

Definition

Architektursicht (View): Repräsentation eines ganzen Systems aus der Perspektive einer kohärenten Menge von Anliegen (IEEE P1471 2002).

Definition

Architekturblickwinkel (Viewpoint): Spezifikation der Regeln und Konventionen, um eine Architektursicht zu konstruieren und zu benutzen (IEEE P1471 2002).

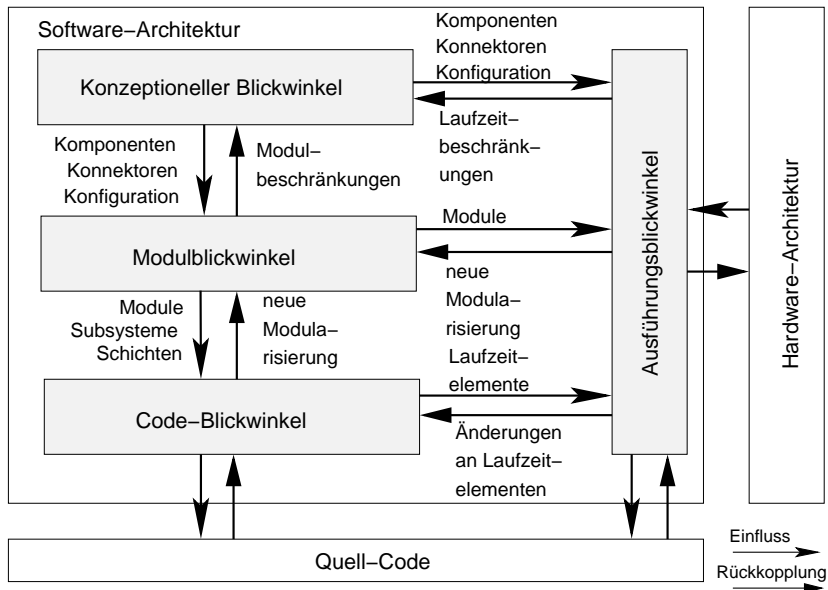
Ein Blickwinkel ist ein Muster oder eine Vorlage, von der aus individuelle Sichten entwickelt werden können, durch Festlegung von

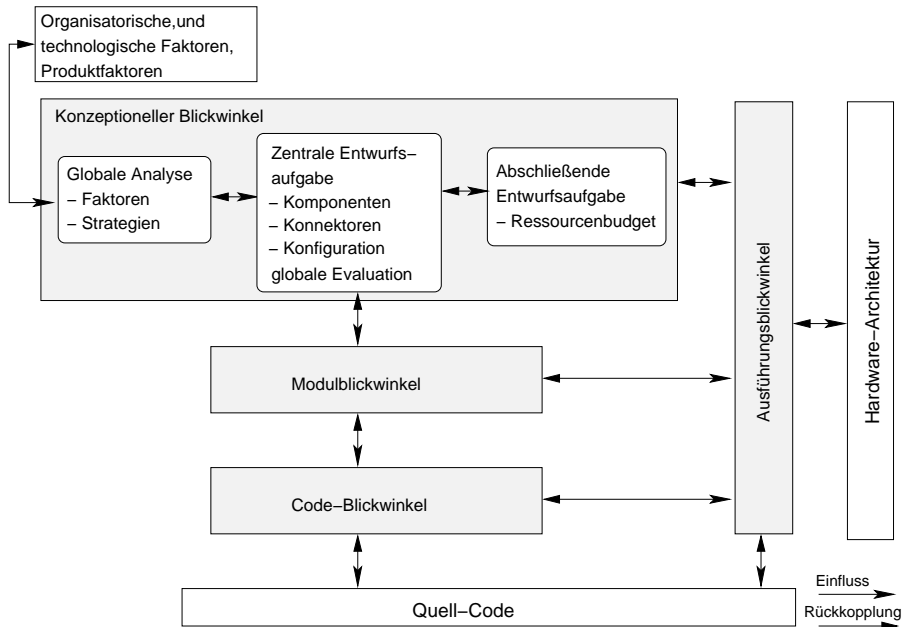
- Zweck,
- adressierte Betrachter,
- und Techniken für Erstellung, Gebrauch und Analyse.

Unterschiedliche Sichten helfen der Strukturierung: Separation of Concerns.

- **Konzeptioneller Blickwinkel:** beschreibt logische Struktur des Systems; abstrahiert weitgehend von technologischen Details
- **Modulblickwinkel:** beschreibt die statische logische Struktur des Systems
- **Ausführungsblickwinkel:** beschreibt die dynamische logische Struktur des Systems
- **Code-Blickwinkel:** beschreibt die „anfassbaren“ Elemente des Systems (Quelldateien, Bibliotheken, ausführbare Dateien etc.)

Siemens-Blickwinkel (Hofmeister u. a. 2000)





Konzeptioneller Blickwinkel (Hofmeister u. a. 2000)

- ist der Anwendungsdomäne am nächsten
- Systemfunktionalität wird abgebildet auf
 - **Konzeptionelle Komponenten**: rechenbetonte Elemente oder Datenhaltung
 - **Konnektoren**: Kontroll- und Datenfluss zwischen konzeptionellen Komponenten

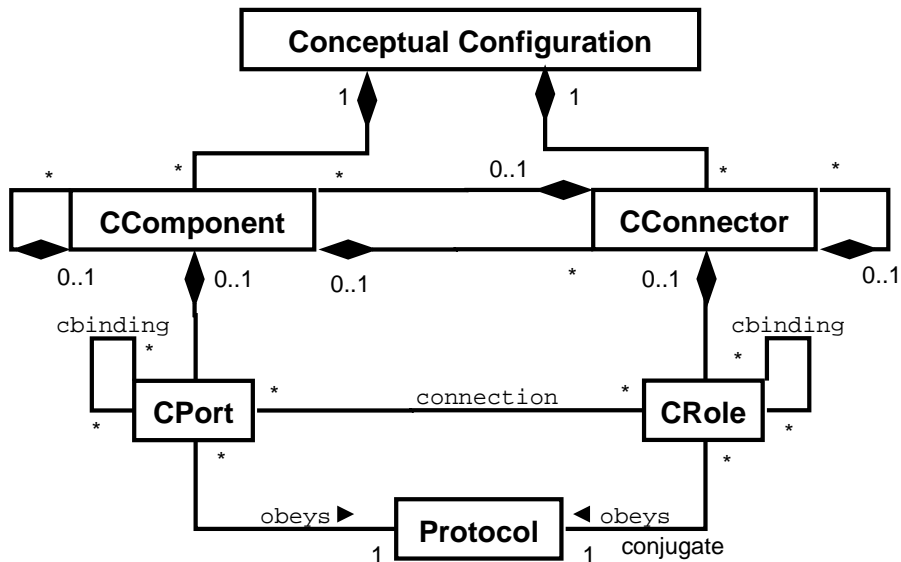
Konzeptioneller Blickwinkel (Hofmeister u. a. 2000)

- ist der Anwendungsdomäne am nächsten
- Systemfunktionalität wird abgebildet auf
 - **Konzeptionelle Komponenten**: rechenbetonte Elemente oder Datenhaltung
 - **Konnektoren**: Kontroll- und Datenfluss zwischen konzeptionellen Komponenten

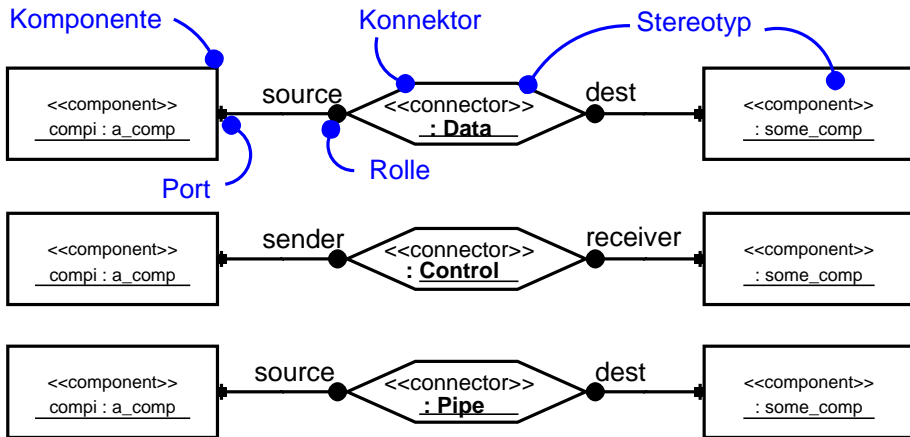
Engineering-Belange:

- Wie erfüllt das System seine Anforderungen?
- Wie werden Commercial-off-the-Shelf-Components (COTS) in das System integriert?
- Wie wird domänenspezifische Soft- und Hardware einbezogen?
- Wie kann die Auswirkung von Anforderungen oder der Anwendungsdomäne minimiert werden?

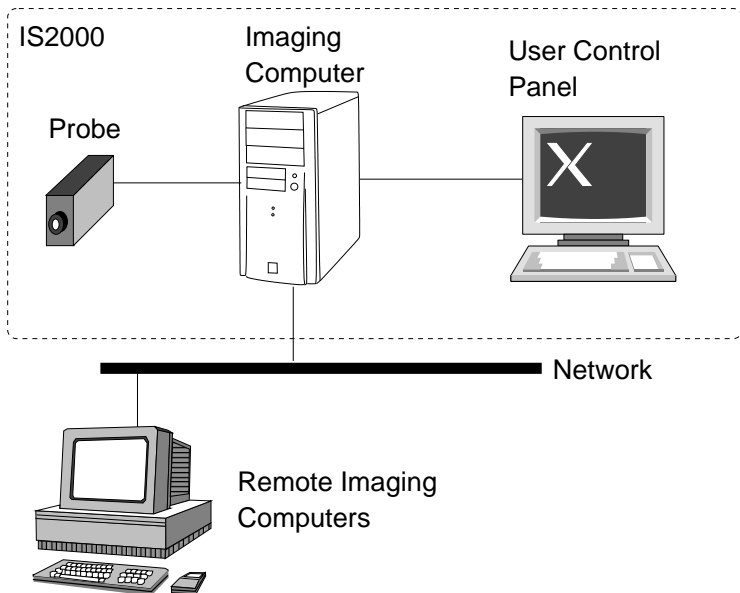
Konzeptioneller Blickwinkel (Hofmeister u. a. 2000)



Komponenten und Konnektoren

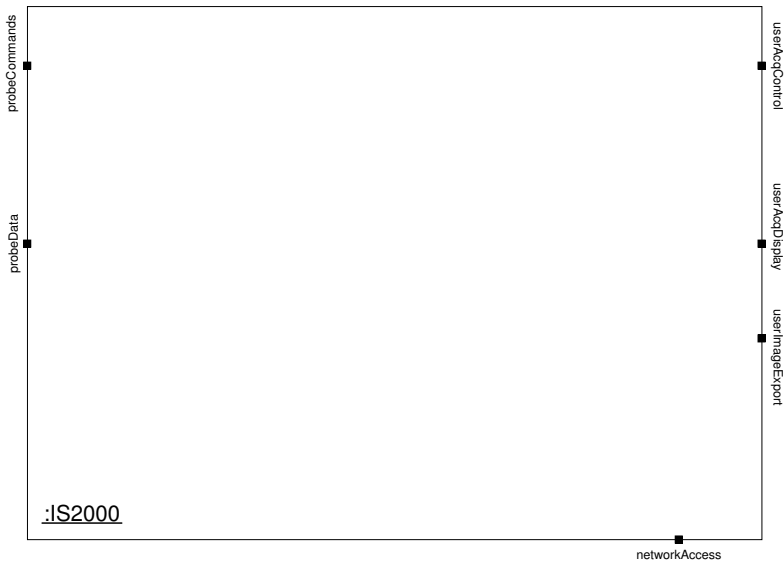


Beispielsystem IS2000 (Hofmeister u. a. 2000)



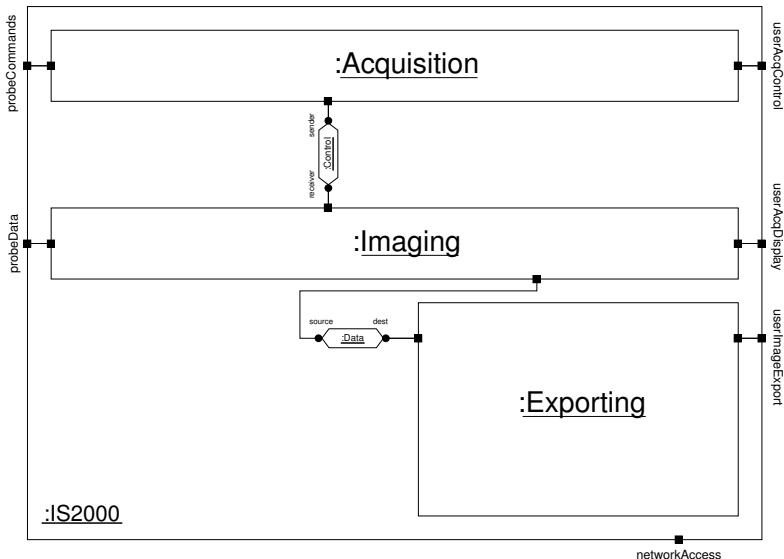
Strategie

Lege externe Schnittstellen fest.



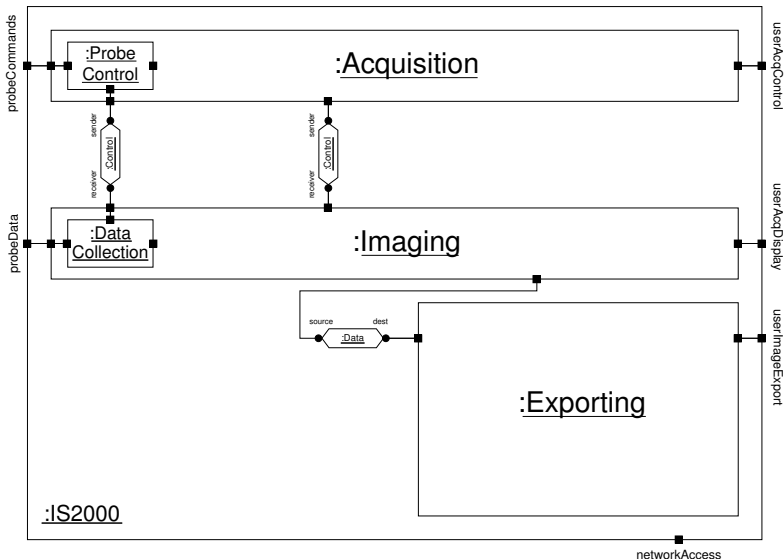
Strategie

Führe Komponenten für Aufzeichnung, Bearbeitung und Export ein.



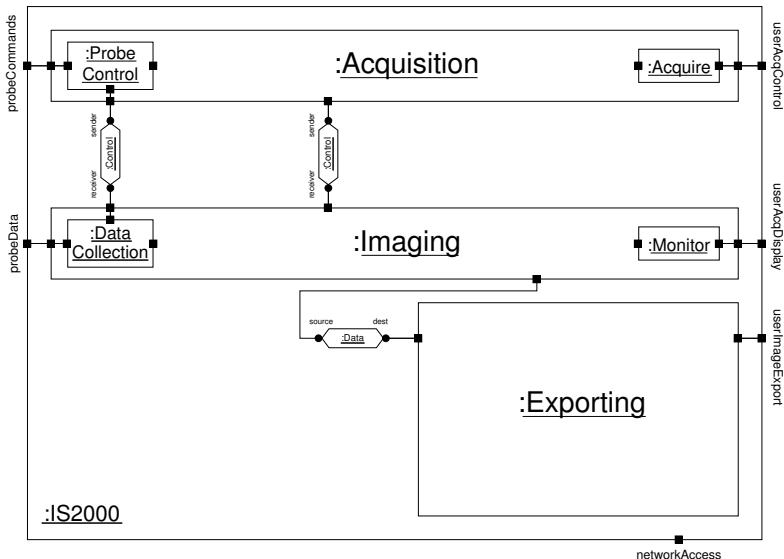
Strategie

Kapselung domänenspezifischer Hardware.



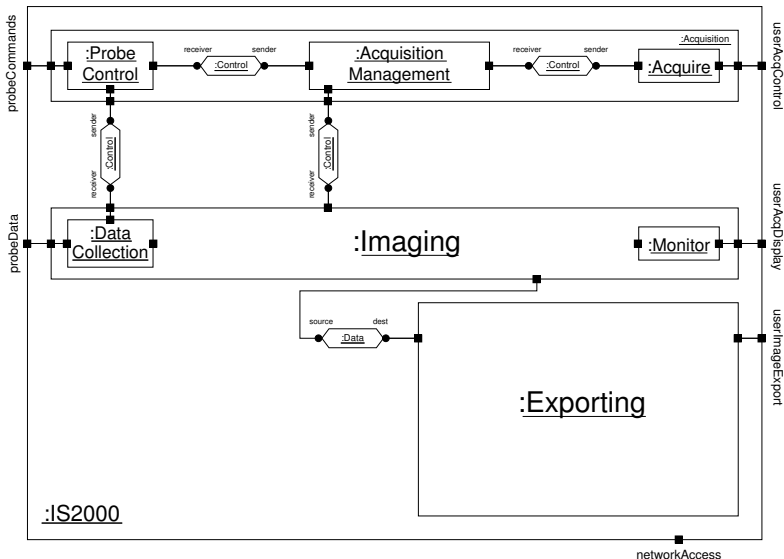
Strategie

Entkopple Interaktionsmodell.



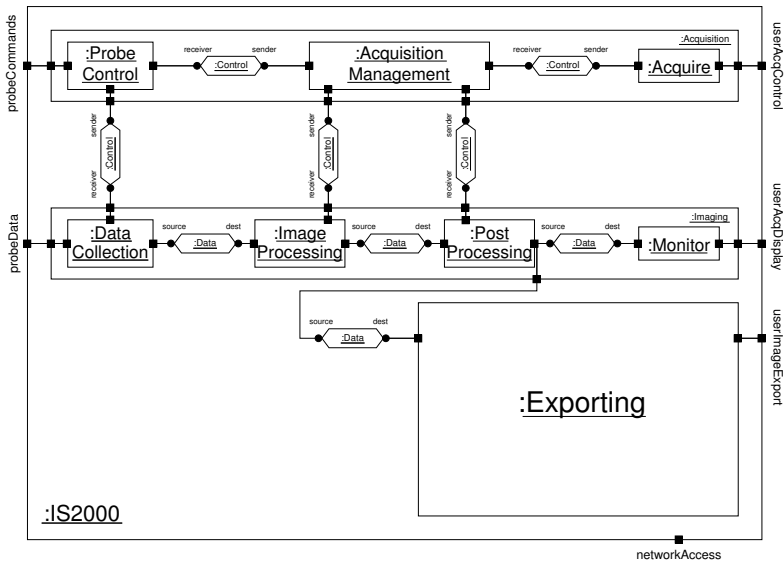
Strategie

Separiere nach Anliegen: Separation of Concern.



Strategie

Separiere zeitkritische Komponenten.



Strategie

Kapselung domänenspezifischer Bild-Daten.

