



UML-Semantik von CIP-Modellen

Das CIP-Metamodell ist in CIP Tool® als Komponenten-Framework realisiert. Das Framework besteht aus Komponentenklassen für Modellkonstrukte. Jede Komponentenklasse definiert, aus welchen Komponenten die Instanzen der Klasse bestehen, und mit welchen Komponenten sie verbunden werden können.

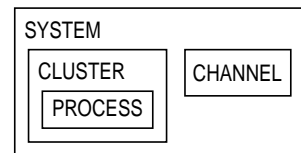
UML eignet sich gut zur Beschreibung der kompositionellen und relationalen Struktur von CIP-Konstrukten. Die CIP-Komponentenklassen werden dazu als UML-Stereotypen deklariert. Stereotypen sind Objektklassen, mit denen das UML-Metamodell für spezifische Bedürfnisse erweitert werden kann.

Die grafischen CIP-Konstrukte von CIP Tool stellen bestimmte Kompositionen und Verknüpfungen von Modellkomponenten dar. Im Folgenden wird die Komponentenarchitektur dieser Konstrukte mit UML-Klassendiagrammen dargestellt.

CIP - Architektur kooperierender Komponenten

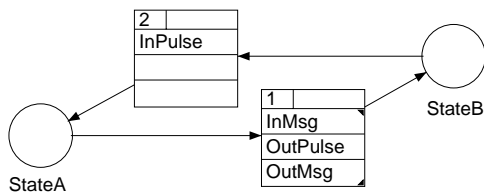
Ein System setzt sich aus Clustern zusammen, die aus synchron interagierenden Prozessen bestehen. Prozesse werden als erweiterte Zustandsmaschinen spezifiziert. Die Cluster sind asynchron kooperierende Subsysteme, die verteilt implementiert werden können. Die Prozesse der verschiedenen Cluster kommunizieren asynchron über aktive Kanäle.

UML Klassendiagramm



Prozesse und Kanäle

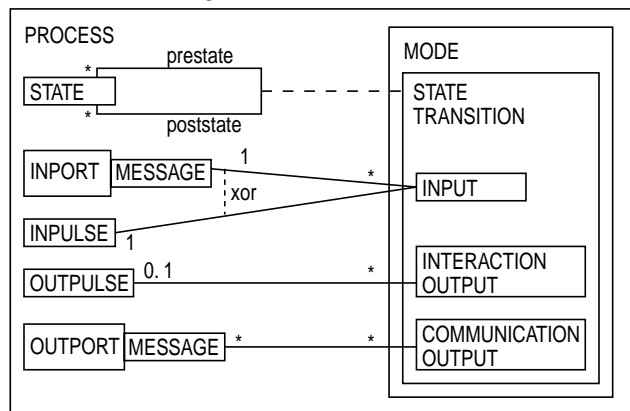
PROCESS - Erweiterte Zustandsmaschine



Transitionsstruktur eines Prozessmodus

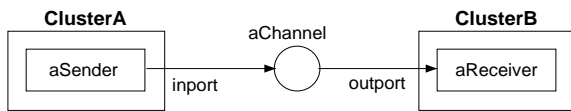
Bei Prozessen werden zwei Kategorien von Input- und Outputsymbolen unterschieden: *Pulse* sind Signale für die synchrone Interaktion innerhalb eines Clusters. *Messages* werden asynchron mittels Kanälen übertragen. Ein Prozess kann mehrere alternative Modi enthalten. Die Erweiterung mit primitiven Komponenten (C/C++ Konstrukte) wie Variablen und Operationen ist hier nicht dargestellt.

UML Klassendiagramm



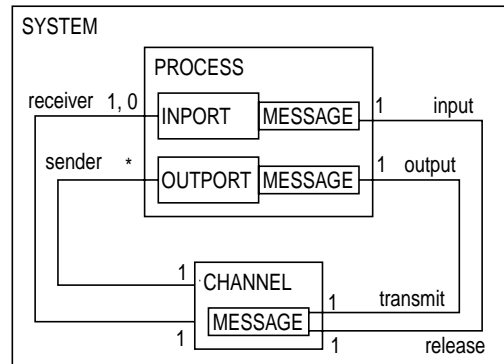
Transitionsstruktur eines Prozesses

COMMUNICATION



Kommunikation zweier Prozesse

Asynchrone Kommunikation bedeutet, dass das Senden und Empfangen einer Meldung nicht zur gleichen Zeit stattfindet.



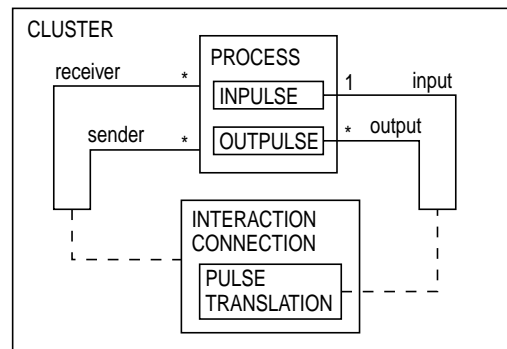
Kommunikationsverbindung

INTERACTION



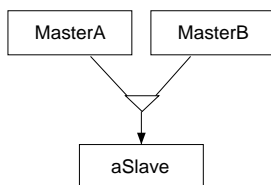
Interaktion zweier Prozesse

Zur Spezifikation einer Interaktionsverbindung gehört eine funktionale Assoziation der Sender-Outputpulse mit Empfänger-Inputpulsen. Diese Funktion wird als *Pulse-Translation* bezeichnet.



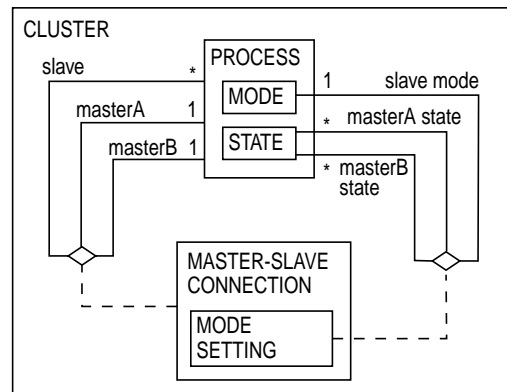
Interaktionsverbindung

MASTER-SLAVE-Hierarchie



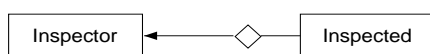
Ein Slaveprozess mit zwei Masterprozessen

Für jede Master-Slave-Assoziation wird in einer Tabelle definiert (*Modesetting*), wie die Zustände der Masterprozesse den aktiven Modus des Slaveprozesses bestimmen.



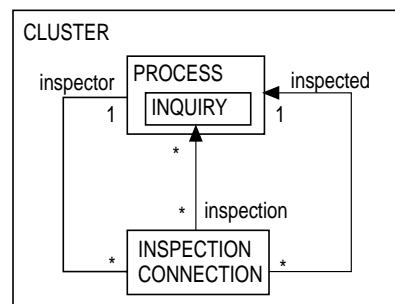
Master-Slave-Verbindung

STATE INSPECTION



Inspektion eines Prozesses

Über Abfragefunktionen (*Inquiries*) liefert ein inspizierter Prozess Information, die von seinem Zustand und seinen Instanzvariablen abhängt.



Inspektionsverbindung