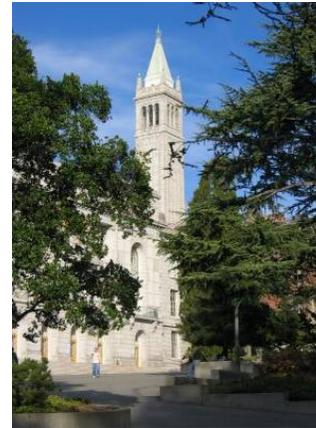


Ptolemy

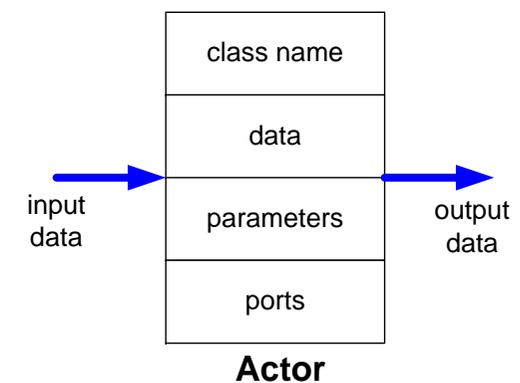
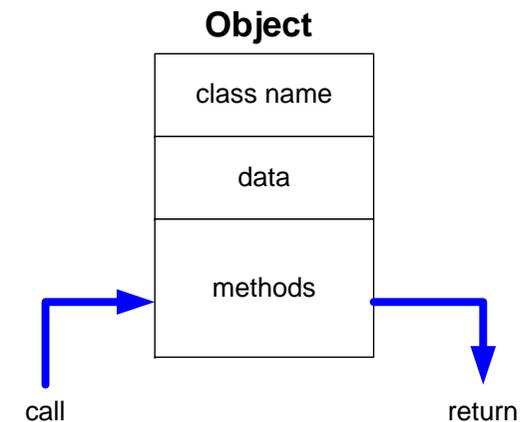
- Das Ptolemy*-Projekt an der UC Berkeley untersucht verschiedene Modellierungsmethodiken für eingebettete Systeme mit einem Fokus auf verschiedene Ausführungsmodelle (Models of Computation)
- Ptolemy unterstützt
 - Modellierung
 - Simulation
 - Codegenerierung
 - Formale Verifikation (teilweise)
- Weitere Informationen unter: <http://ptolemy.eecs.berkeley.edu/>



***Claudius Ptolemaeus**, (* um 100, vermutlich in Ptolemais Hermii, Ägypten; † um 175, vermutlich in Alexandria), war ein griechischer Mathematiker, Geograph, Astronom, Astrologe, Musiktheoretiker und Philosoph. Ptolemäus schrieb die *Mathematike Syntaxis* („mathematische Zusammenstellung“), später *Megiste Syntaxis* („größte Zusammenstellung“), heute *Almagest* (abgeleitet vom Arabischen *al-Majisṭī*) genannte Abhandlung zur Mathematik und Astronomie in 13 Büchern. Sie war bis zum Ende des Mittelalters ein Standardwerk der Astronomie und enthielt neben einem ausführlichen Sternenkatalog eine Verfeinerung des von Hipparchos von Nicäa vorgeschlagenen geozentrischen Weltbildes, das später nach ihm *Ptolemäisches Weltbild* genannt wurde. (Wikipedia)

Ptolemy: Aktororientiertes Design

- Ptolemy-Modelle basieren auf Aktoren anstelle von Objekten
- Objekte:
 - Fokus liegt auf Kontrollfluss
 - Objekte werden manipuliert
- Aktoren
 - Fokus liegt auf Datenfluss
 - Aktoren manipulieren das System
- Vorteil beider Ansätze: erhöhte Wiederverwendbarkeit
- Vorteil von Aktoren: leichtere Darstellung von Parallelität



Ptolemy: Aktororientiertes Design

- Ausführungsmodelle (models of computation) bestimmen die Interaktion von Komponenten/Aktoren.
- Die Eignung eines Ausführungsmodells hängt von der Anwendungsdomäne, aber auch der verwendeten Hardware, ab.
- In Ptolemy wird durch die Einführung von „Dirigenten“ (director) die funktionale Ausführung (Verschaltung der Aktoren) von der zeitlichen Ausführung (Abbildung im Direktor) getrennt.
- Aktoren können unter verschiedenen Ausführungsmodellen verwendet werden (z.B. synchron, asynchron)
- Verschiedene Ausführungsmodelle können hierarchisch geschachtelt werden (modal models).
 - Typisches Beispiel: Synchroner Datenfluss und Zustandsautomaten

Example Ptolemy Model of Comp.: Synchronuous Dataflow

- Prinzip:
 - Annahme: unendlich schnelle Maschine
 - Daten werden zyklisch verarbeitet
 - Pro Runde wird genau einmal der Datenfluss ausgeführt
- Vorteile:
 - Statische Speicherallokation
 - Statischer Schedule berechenbar
 - Verklemmungen detektierbar
 - Laufzeit kann bestimmt werden
- Werkzeuge:
 - Matlab
 - Labview
 - **EasyLab**

