Service-Oriented Computing in Recomposable Embedded Systems

Autonomous + Backend Support

Yinong Chen

Department of Computer Science and Engineering

ASU Arizona State University

http://www.public.asu.edu/~ychen10/

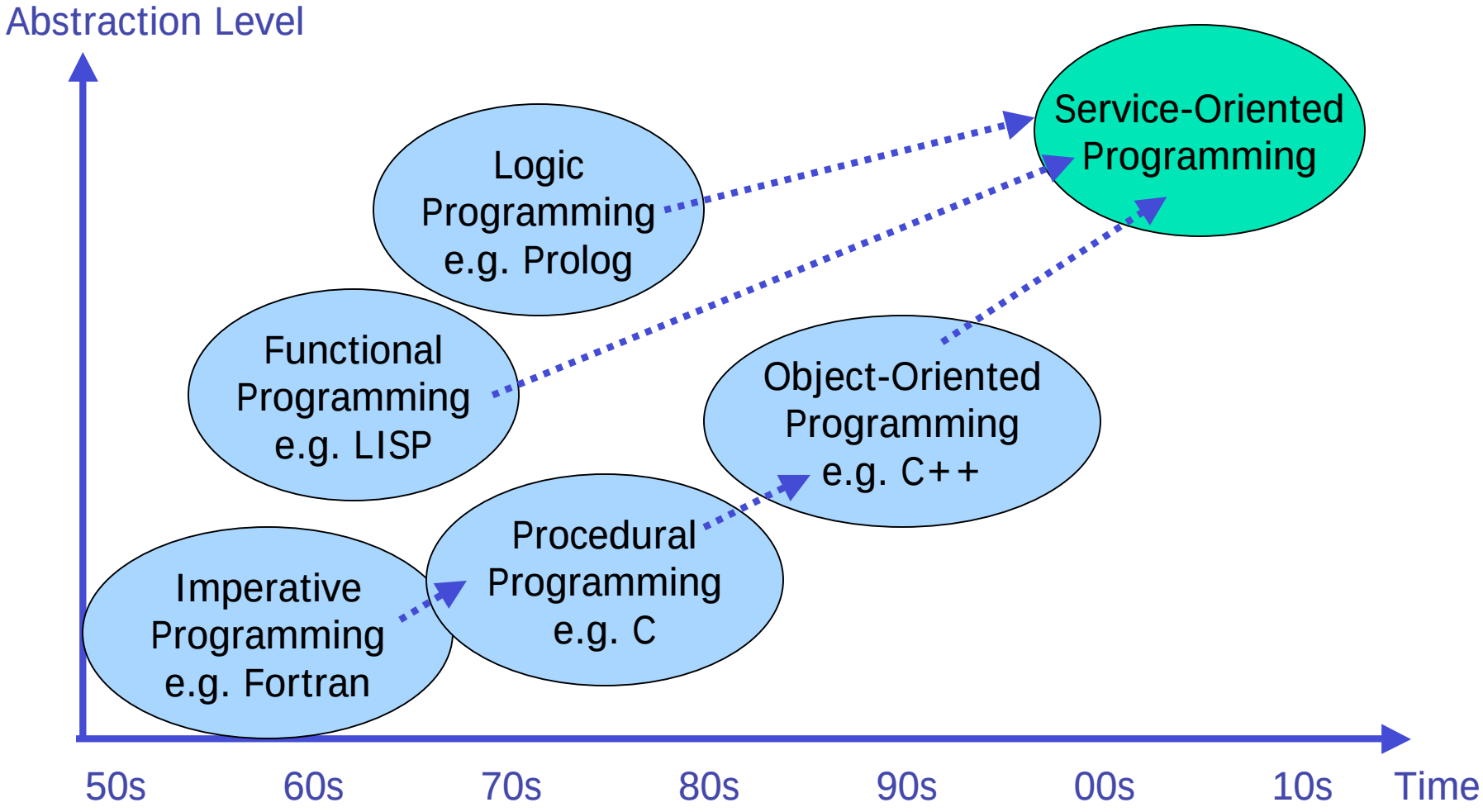
Motivation

- Embedded systems / Robots have limited capacity to carry programs that handle all possible situations;
- Unforeseeable environmental situations can occur;
- Faults can occur and without on-site repair;
- The users want to modify the system (requirements) without stopping the system.

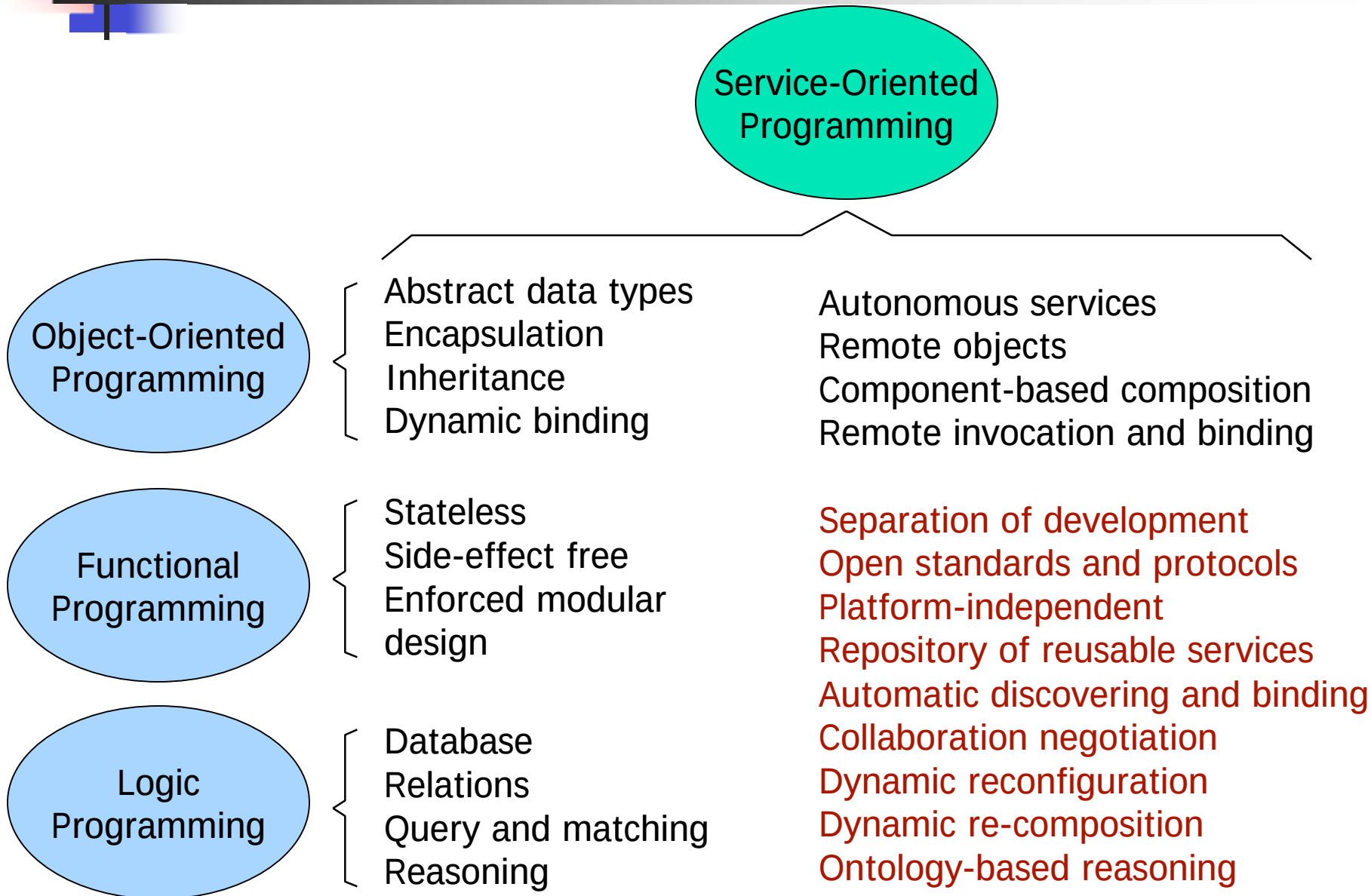
Roadmap

- **Service-Oriented Computing: a New Paradigm**
- Service Providing
- Service Registry and Repository
- Application Building
- Application in Recomposable Embedded Systems

Paradigms of Computing

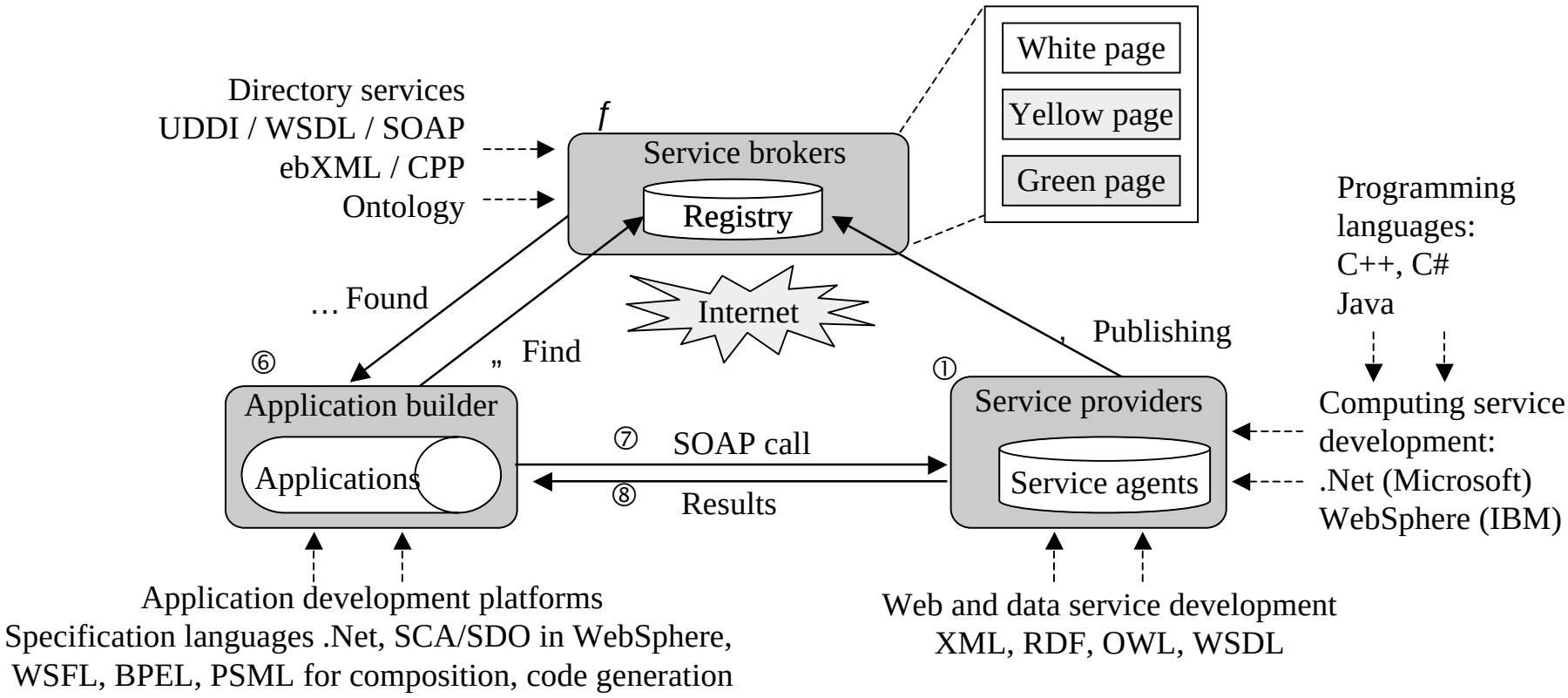


Service-Oriented Computing



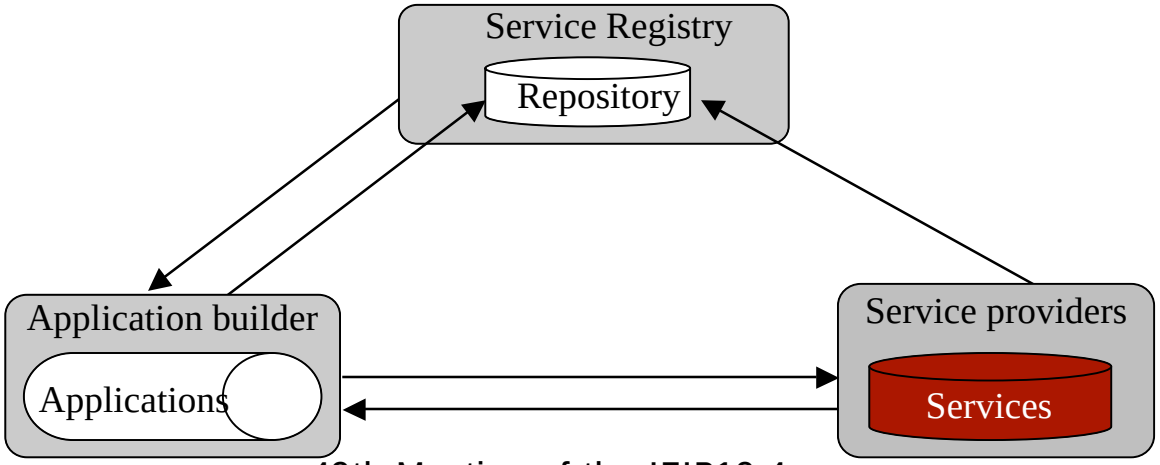
Separation of Development

The Three-Party Model of Service Orientation



Roadmap

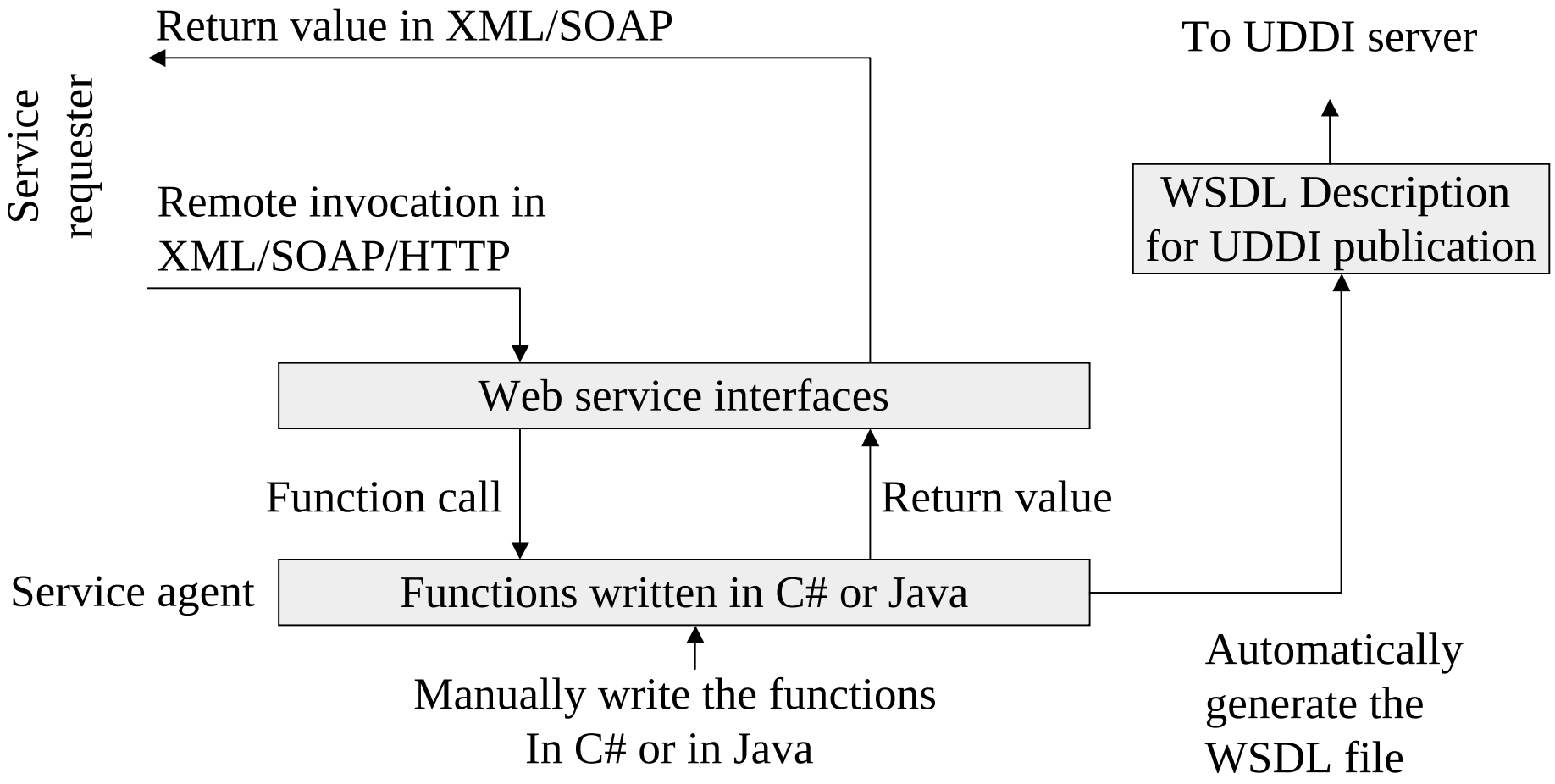
- Service-Oriented Computing: a New Paradigm
- **Service Providing**
- Service Registry and Repository
- Application Building
- Application in Recomposable Embedded Systems



Services Providing (Programming)

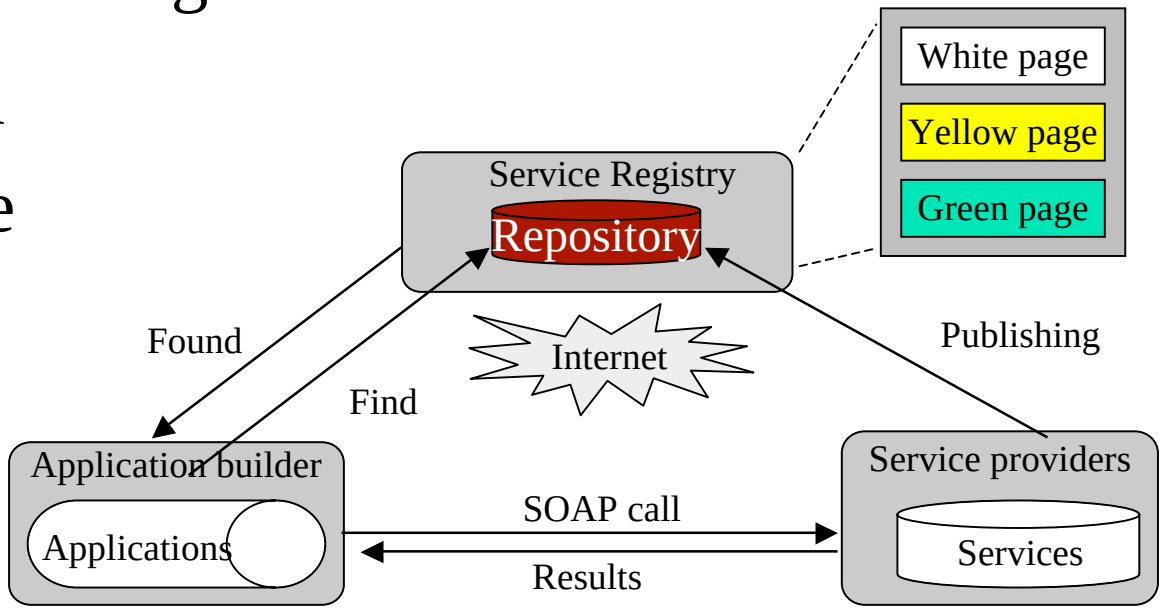
- Services are building blocks of SOC software (COTS)
- Services have an open standard interface, e.g., in WSDL
- Services are placed (published) in an internet-searchable repository
- Services can be automatically discovered (searched)
- Services can be remotely invoked via a standard protocol, e.g., SOAP – remote procedure call
- Services are platform-independent, it can be written in any languages: Java, C#, etc.
- Every piece of program can be wrapped as a service
- In the near future, most services required will be available.
- There is no need to write new services in most cases
- The programming languages likes Java, C#, C++, etc will be less frequently used.

Services are Wrapped Classes



Roadmap

- Service-Oriented Computing: a New Paradigm
- Service Providing
- **Service Registry and Repository**
- Application Building
- Application in Recomposable Embedded Systems



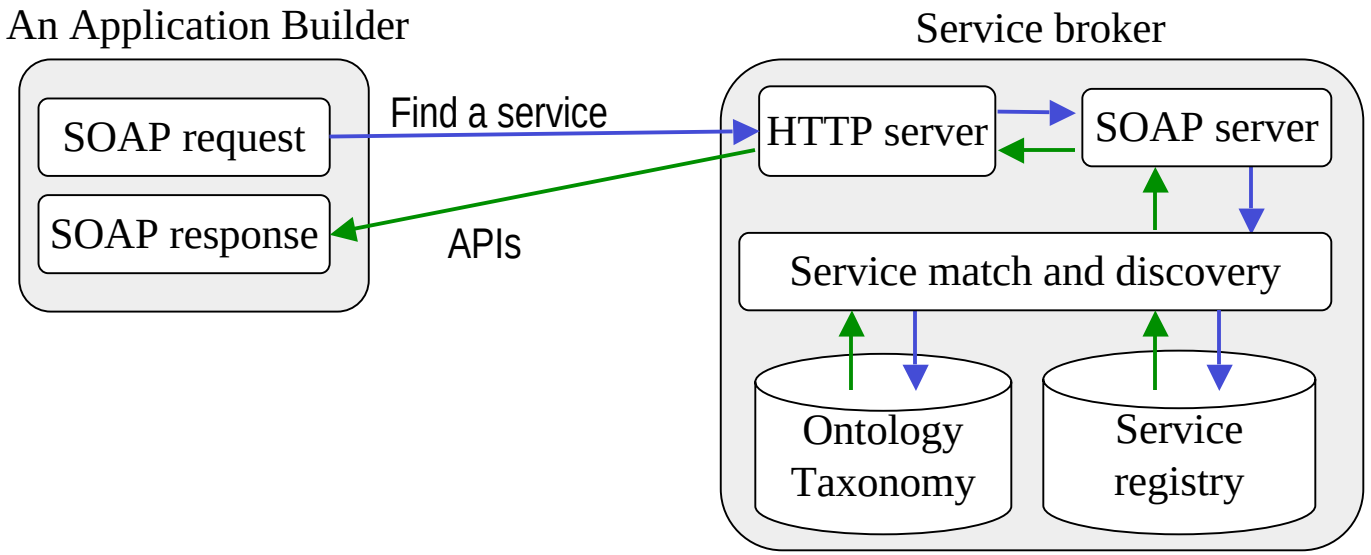
UDDI Service Registry

Universal Description, Discovery, and Integration

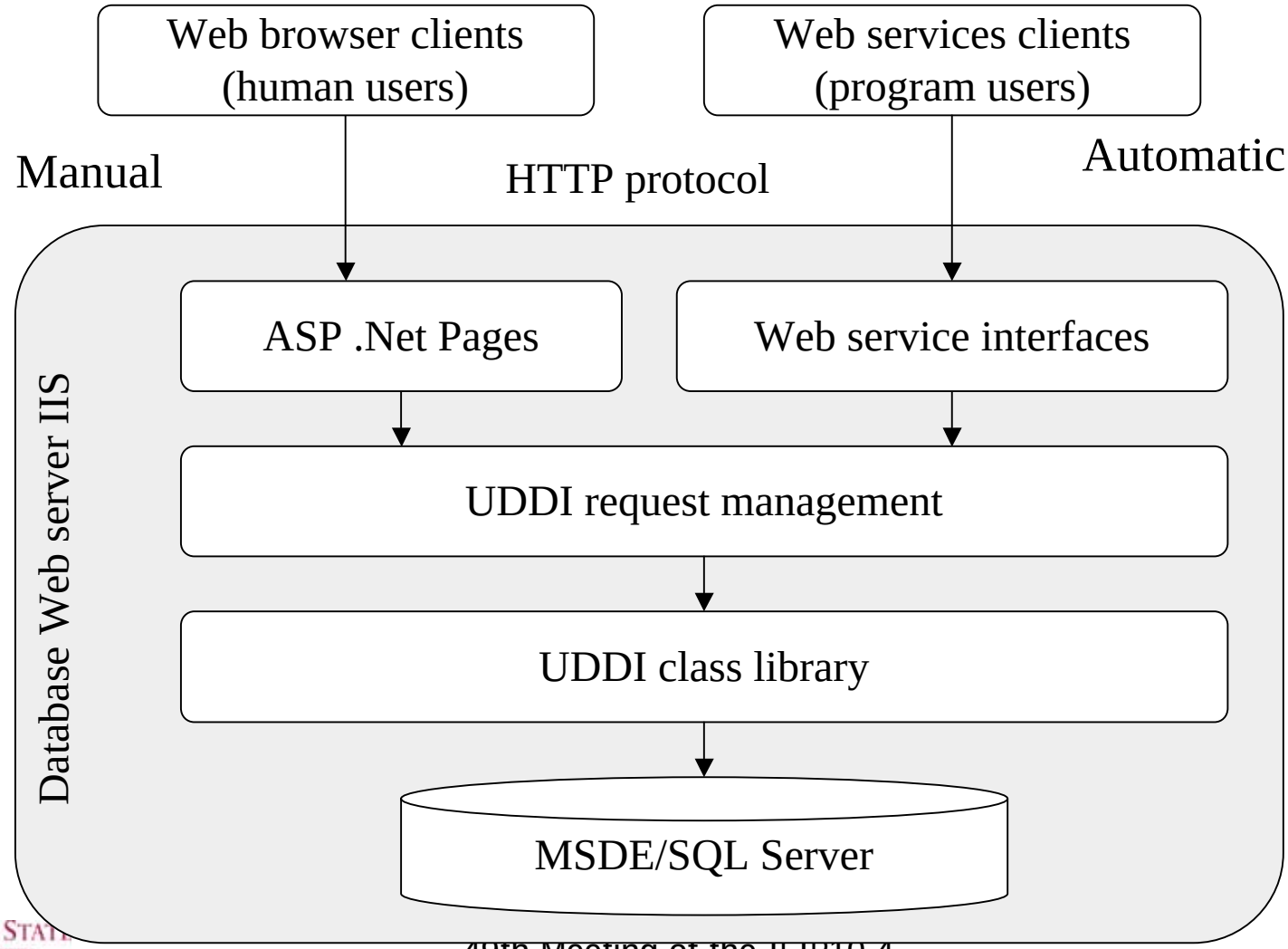
UDDI registry information is organized in three groups:

- **White page** includes service provider's name, identify, e.g., the DUNS number, contact information.
- **Yellow page** includes industry type, product and service type, and geographical location.
- **Green page** includes binding information associated with services, references to the technical models those services implement, and pointers to various file and URL-based discovery mechanisms. The information can be searched and interpreted by programs.

Search and Discover a Service

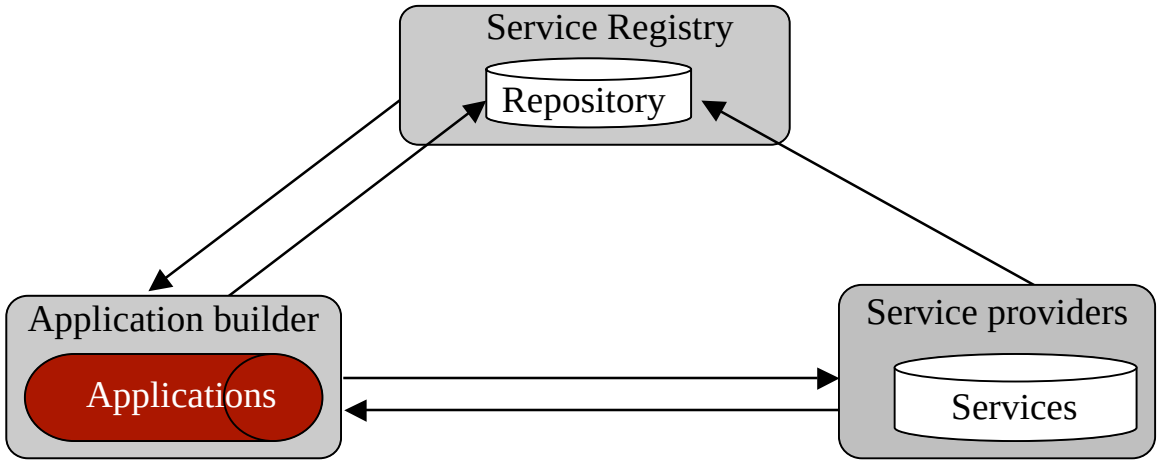


Microsoft Enterprise UDDI Services
http://uddi.microsoft.com/

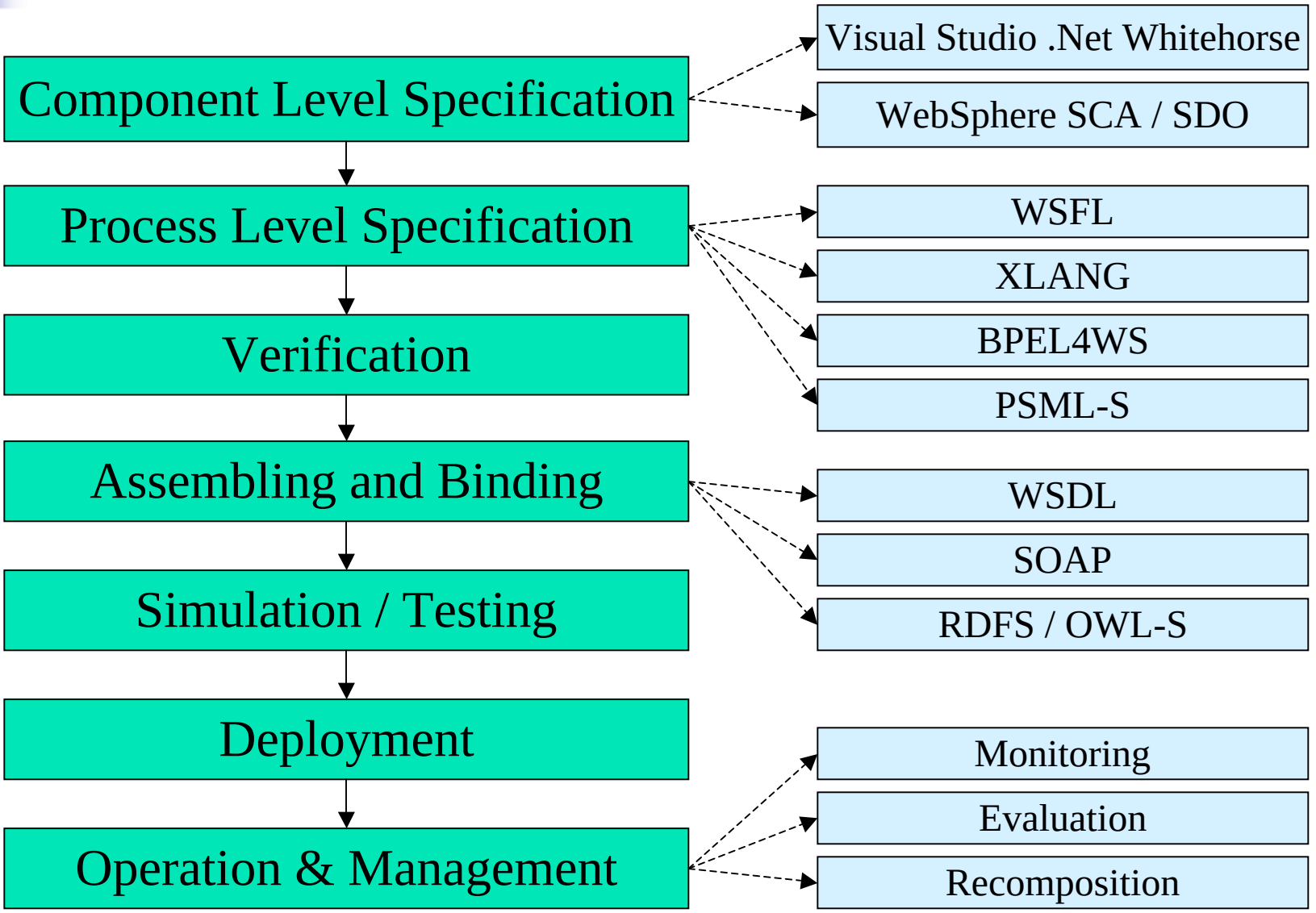


Roadmap

- Service-Oriented Computing: a New Paradigm
- Service Providing
- Service Registry and Repository
- **Application Building**
- Application in Recomposable Embedded Systems



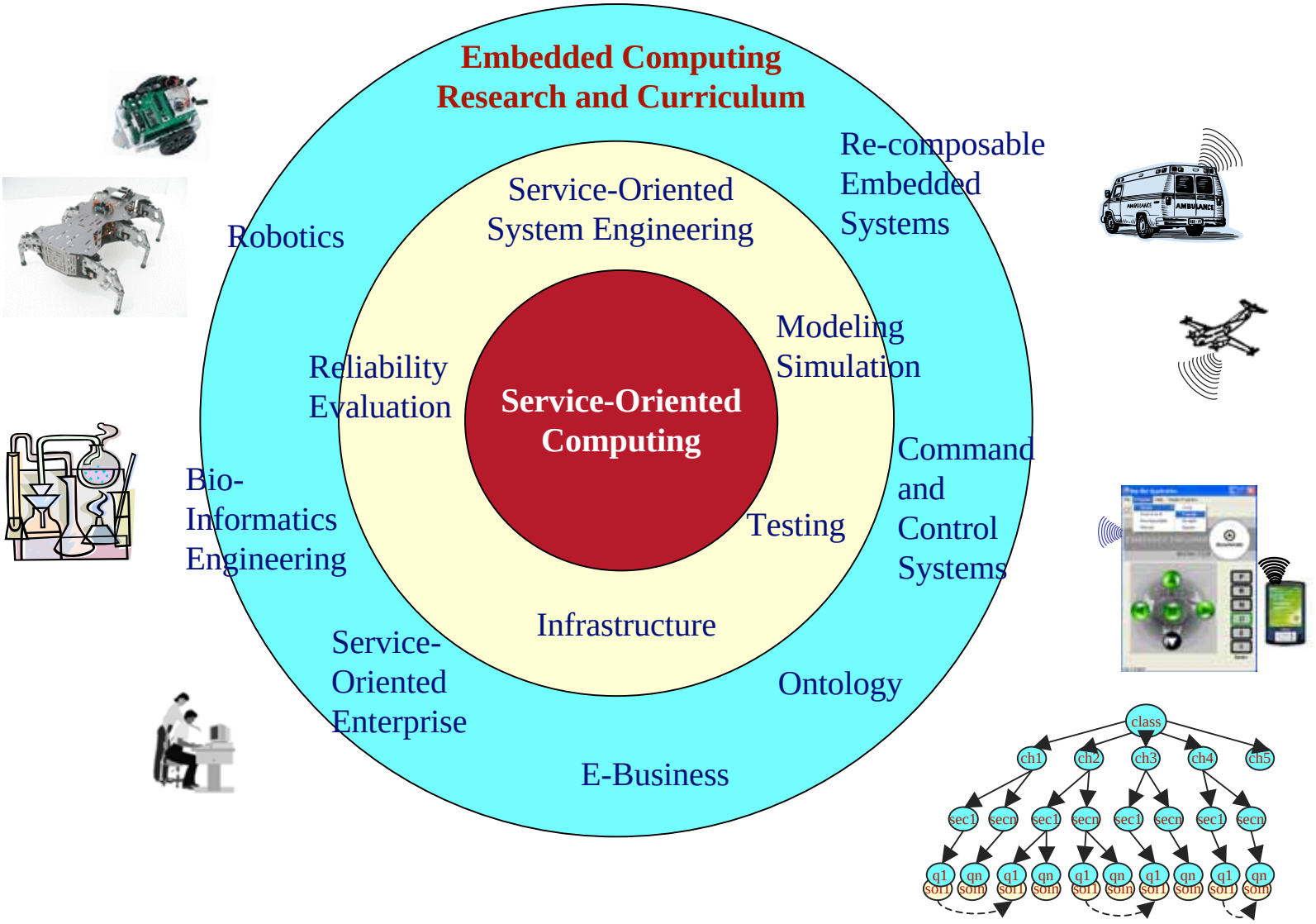
Process of Application Building



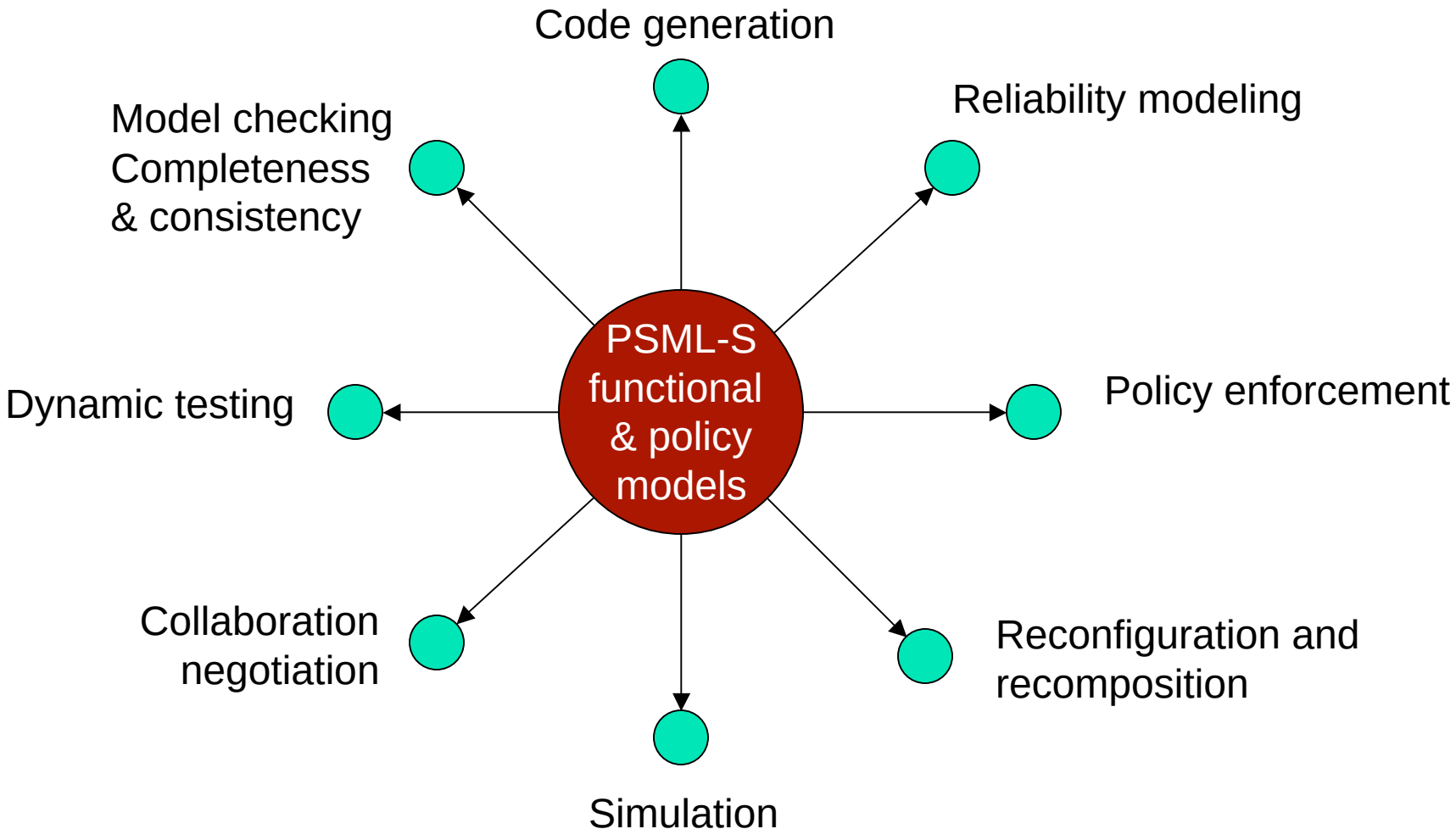
Roadmap

- Service-Oriented Computing: a New Paradigm
- Service Providing
- Service Registry and Repository
- Application Building
- **Application Recomposable Embedded Systems**
  - **Motivation**
  - **Model-driven approach: single model multiple analyses**
  - **Development cycle: A real cycle with a feedback loop**
  - **ASU SO Embedded System Project**

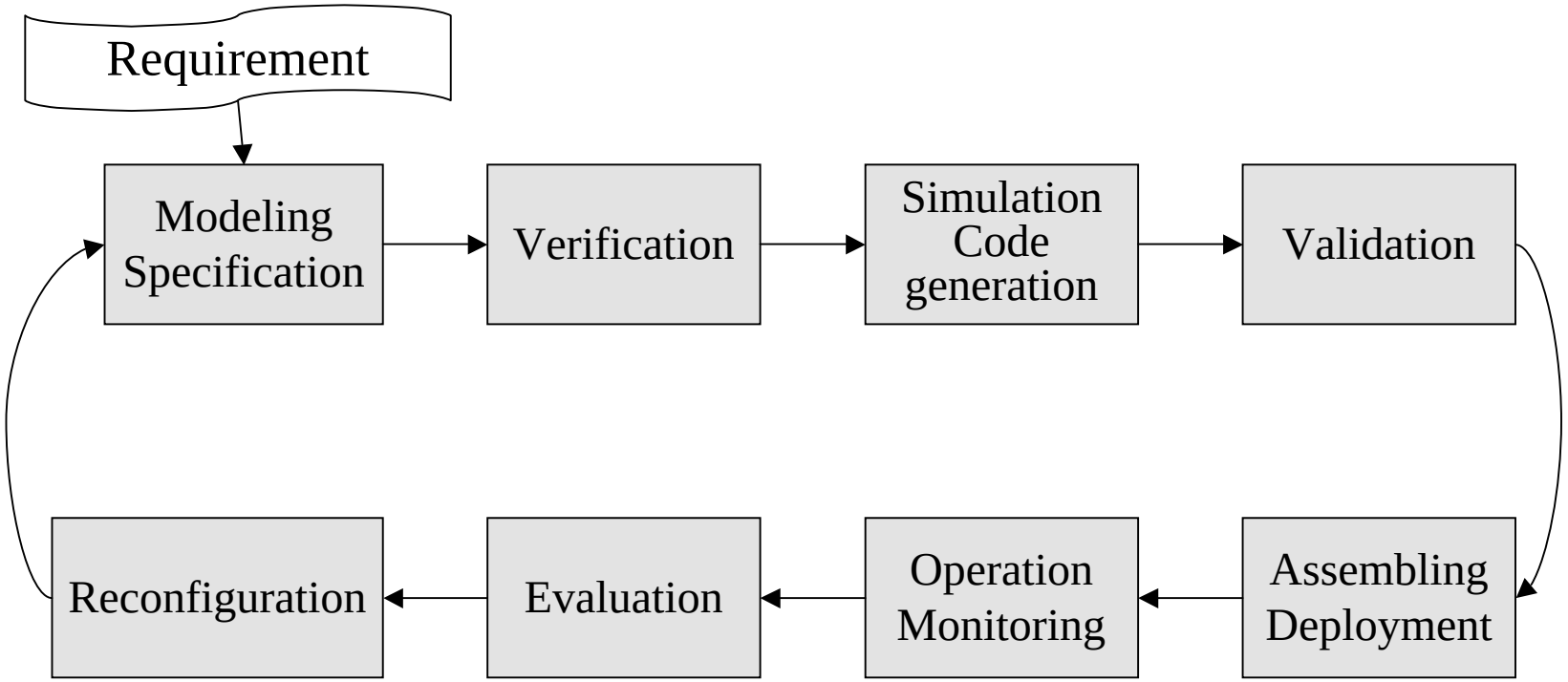
Service-Oriented Computing, System Engineering and Applications at Arizona State University



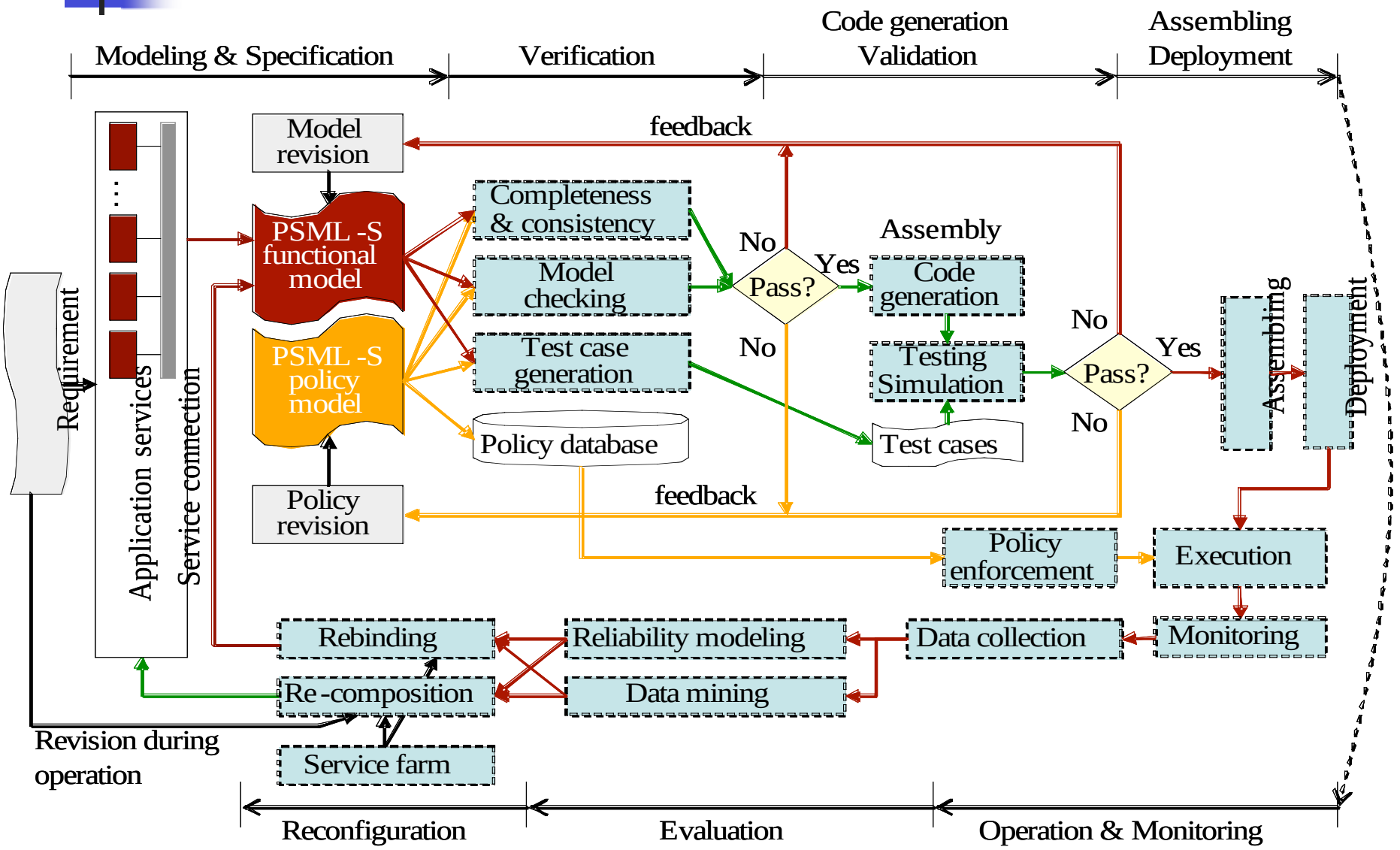
Model-Driven: Single Model Multiple Analyses



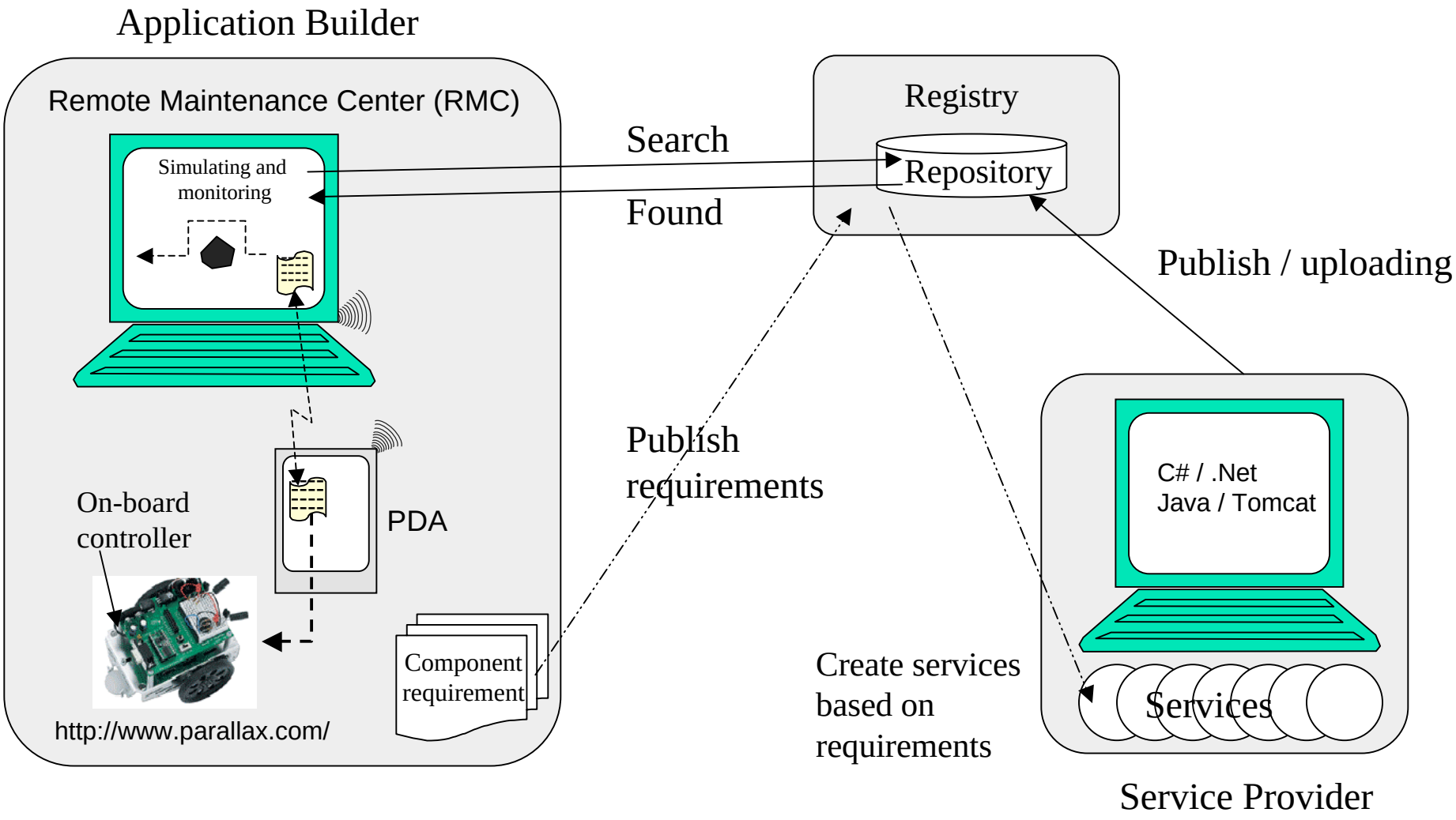
ASU PSML-S based SOC Development Cycle



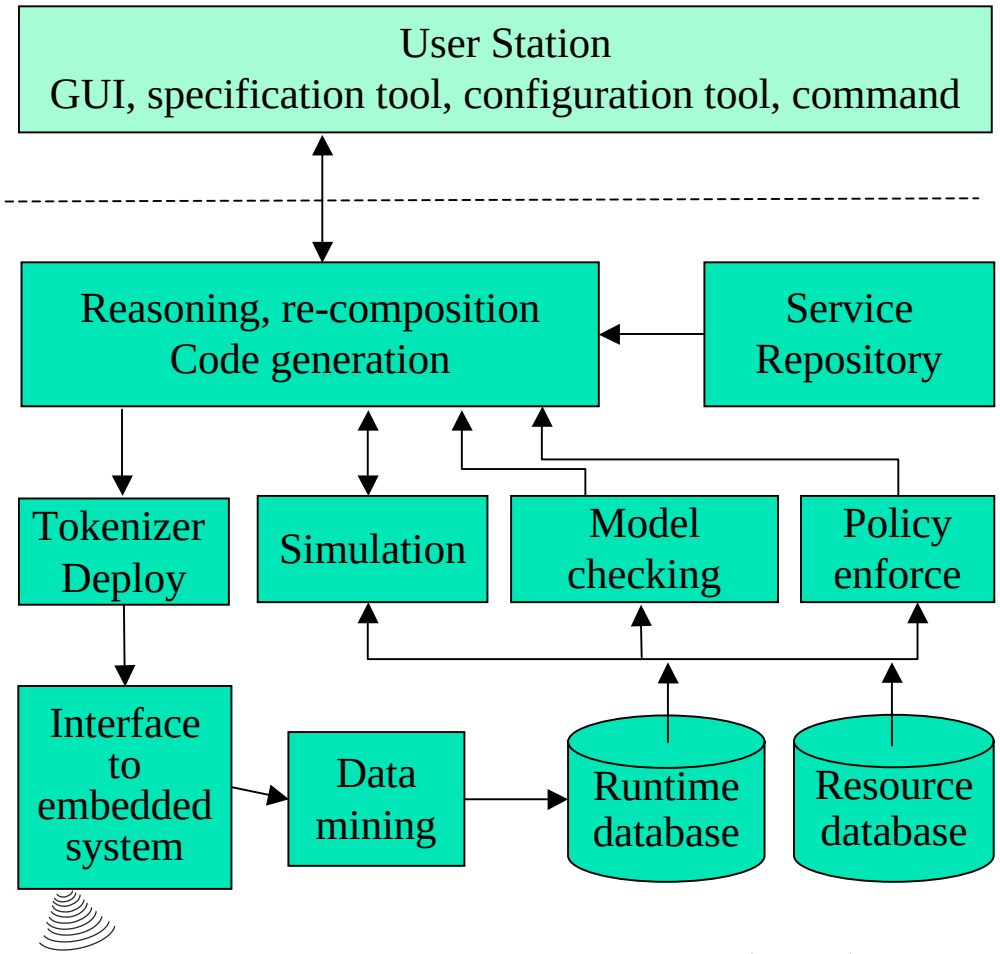
The Development Cycle with More Detail



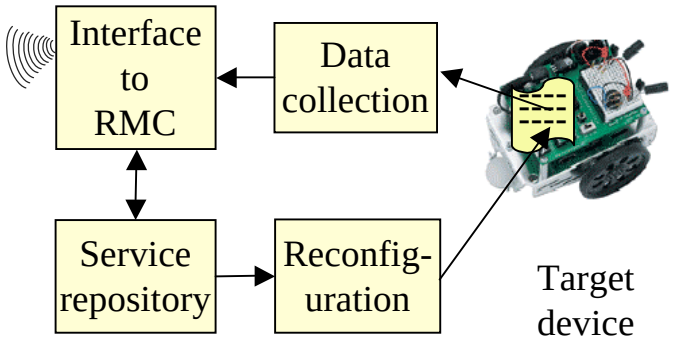
Service-Oriented Recomposable Embedded Systems



System Overview



Remote Maintenance Center (RMC)



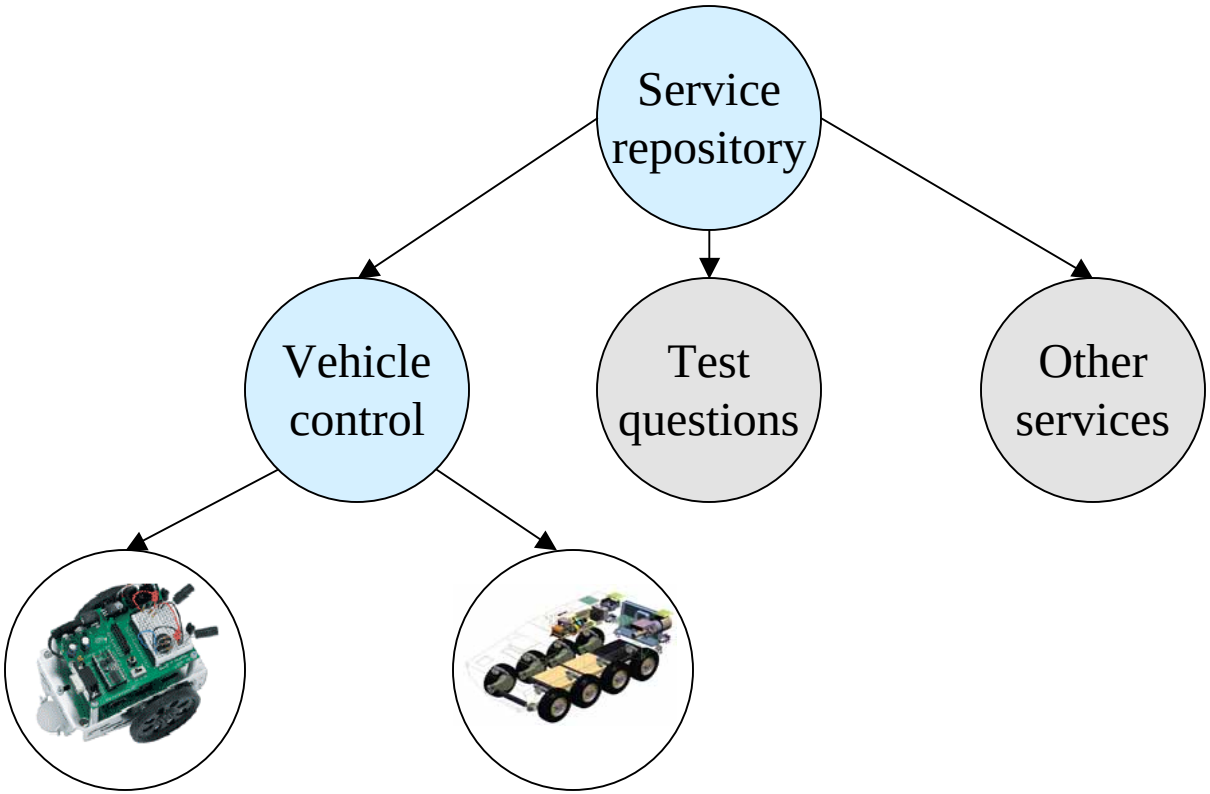
PDA Controller

Service Providing

For each type of vehicle, basic services are defined:

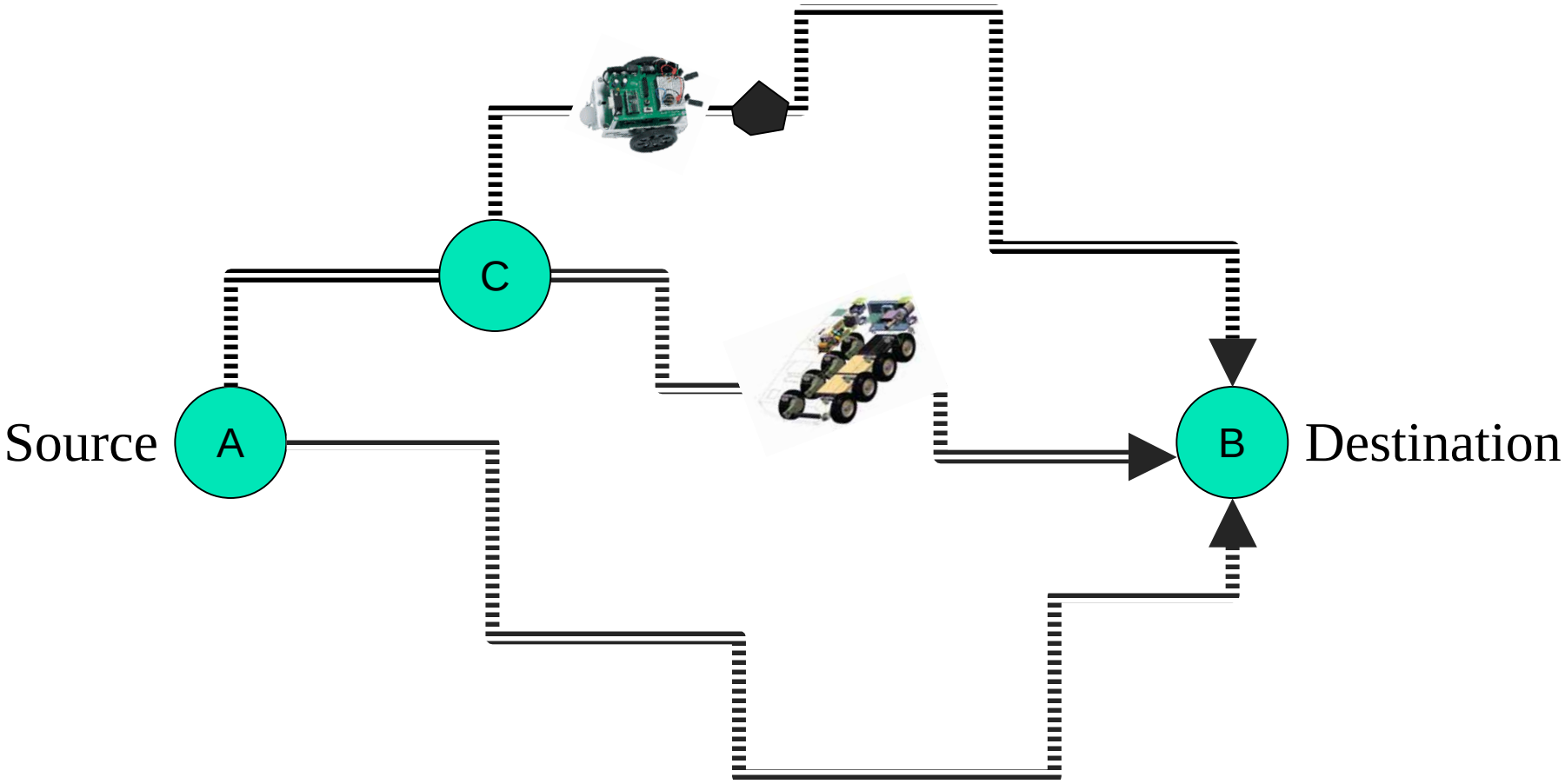
- Forward(n);
- Backward(n);
- TurnLeft(θ); where, θ = 45, 90, 135 degree
- TurnRight(θ); where, θ = 45, 90, 135 degree
- TurnBack(); turn 180 degree
- TurnCircle();
- AutoPath(A, B);
- boolean Obstacle(); return true if there is an obstacle
- …

Service Repository



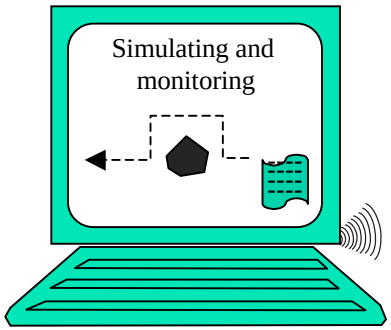
Application Building Based on Services

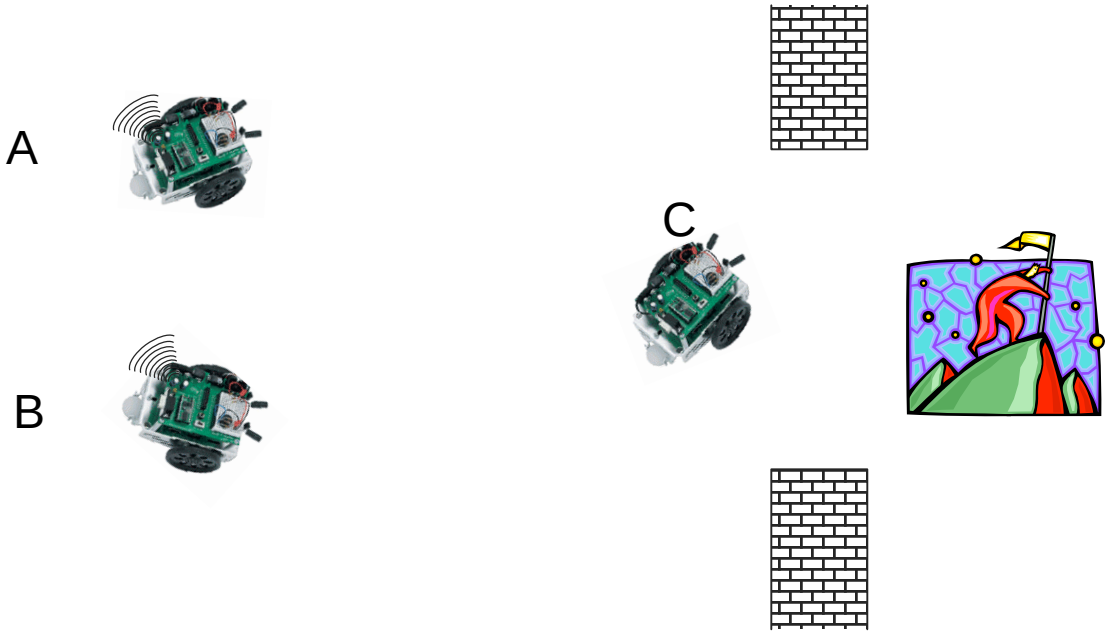
The application logic does not have to be changed when the vehicle is changed.
Replace the component services will have a different vehicle to perform the same tasks.



Application Building Based on Situation

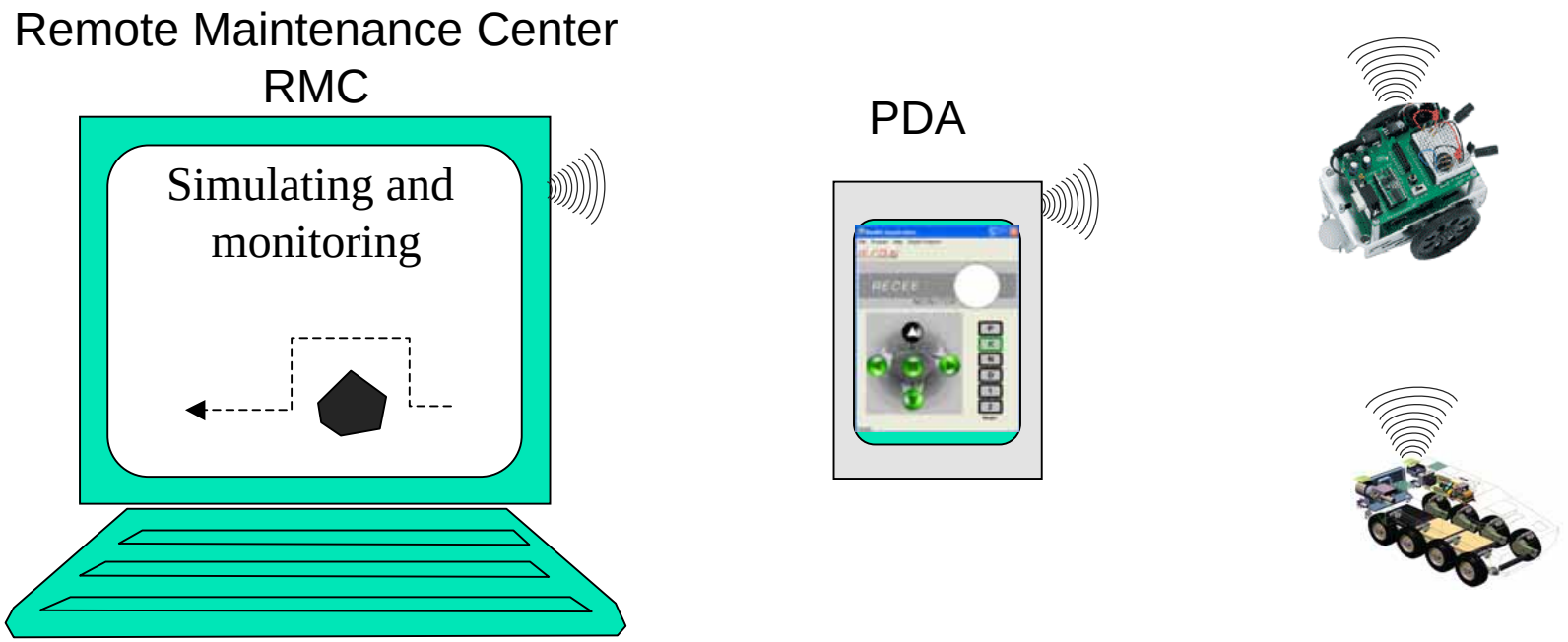
- A and B cooperate to try to move through the gate.
- C tries to block them.
- A and B will test C's intelligence , and send the information back to the RMC
- RMC will construct programs for A and B
- A and B will follow the new program to bypass C.



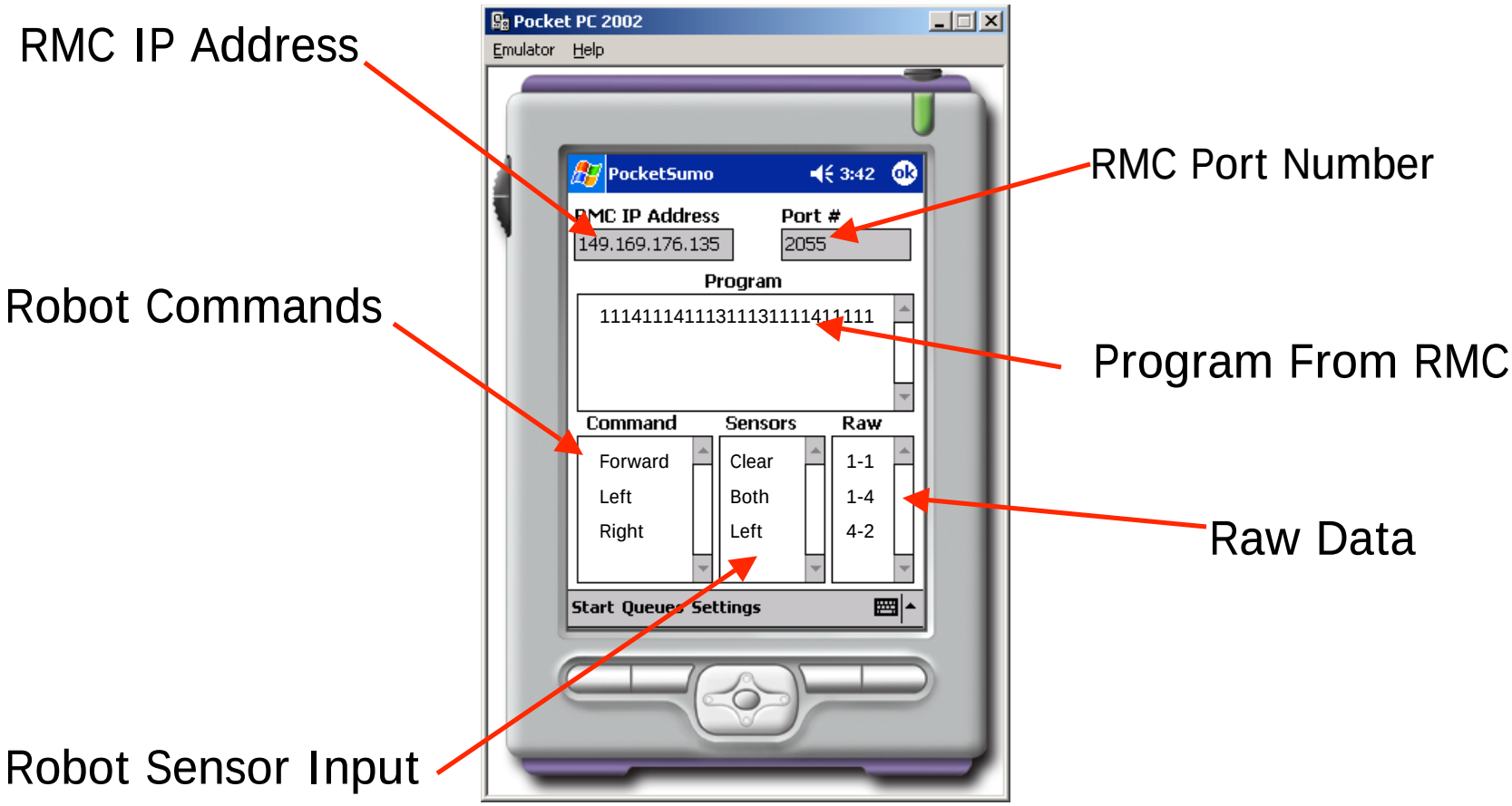


An Optional Layer -- PDA

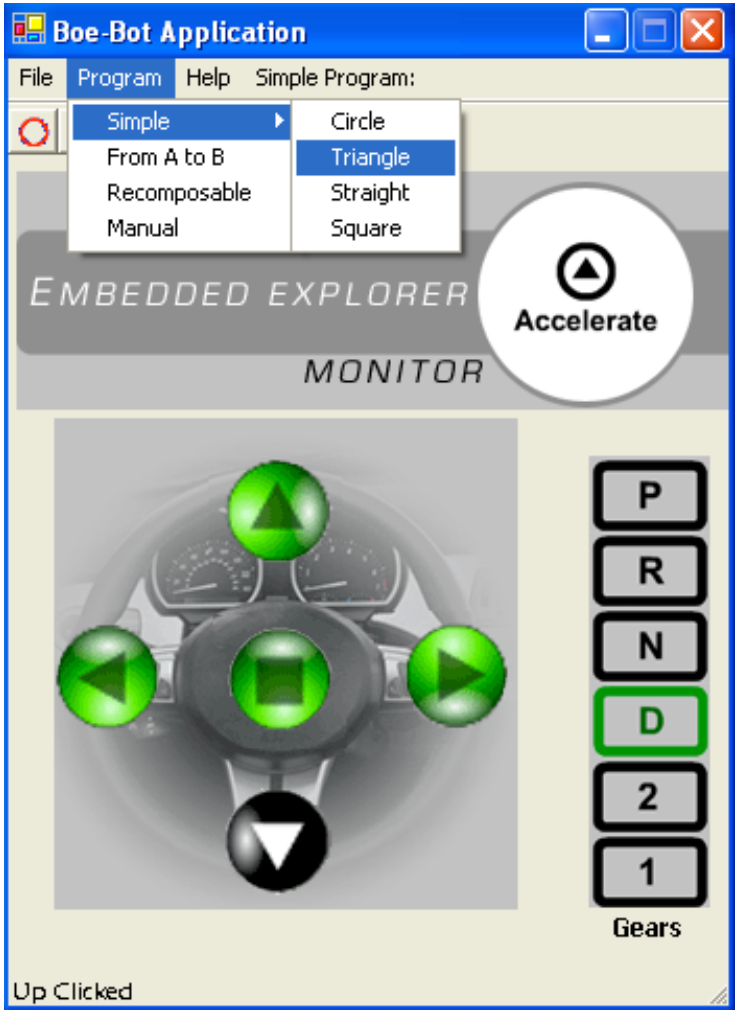
Act as
- a mobile controller of the vehicles
- a more powerful on board computer



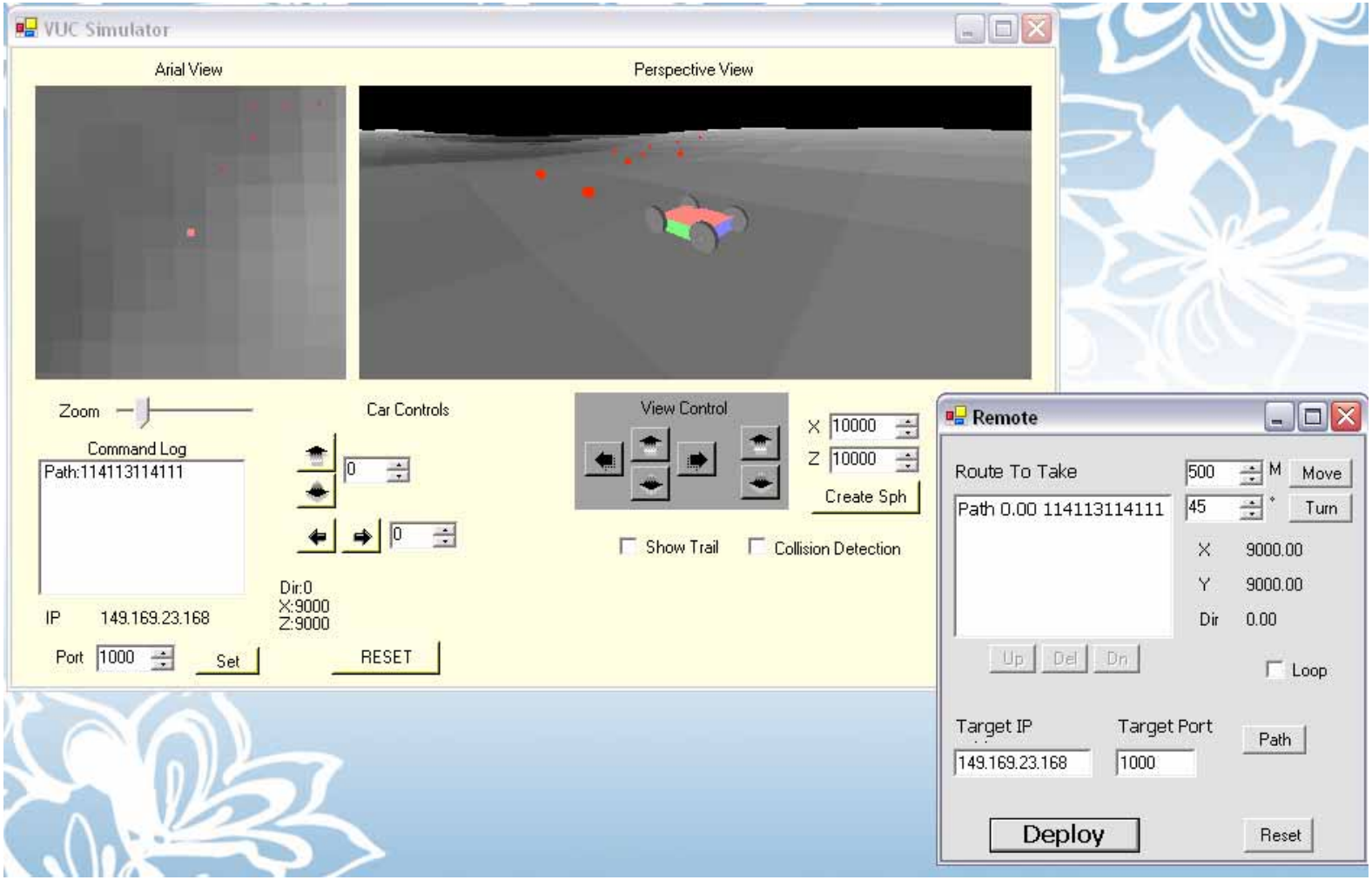
PDA as an Onboard Computer



PDA Manual Mode: Remote Controller



Simulation and Monitoring on RMC



SUMMARY

- Service-Oriented Computing
- Service Providing
- Service Registry and Repository
- Application Building
- Application in Recomposable Embedded Systems
- Demo