ENSE 622 Systems Engineering Requirements, Design, and TradeOff Analysis

Platform-Based Design

Mark Austin

E-mail: austin@isr.umd.edu

Institute for Systems Research, University of Maryland, College Park
1. Tenets of Platform-Based Design
   - Motivation, definition, and benefits.

2. Flow of Design Activities
   - Efficiency and correctness.

3. Meet-me-in-the-Middle Design
   - Architectures and rules for assembly.

4. Platform Stacks
   - Methodology and guidelines for library design.

5. Case Studies. Platform Applications:
   - Engineering Analysis; Design of a Pizza Restaurant; Building Automation Systems; Biomedical Engineering.
Preliminary Observations

- Design is a transformational process that takes a specification and turns it into a product.
- The way in which this process is organized is called a methodology.

Motivation

As systems become progressively more complex, and time-to-market constraints progressively more stringent ...

... the relative cost of systematically exploring design spaces to find good designs, and then verifying and testing behavior will steadily increase unless new approaches are developed.
Definition

We define platform-based design as:

... the creation of a stable architecture that can be rapidly extended, customized for a range of applications (instead of a single product), and delivered to customers for quick deployment.

Benefits

Platform-based design methodologies

... improve the efficiency, correctness and economics of design by restricting the space of design options to pre-defined components, connectors, and rules for assembly (all contained in a library).
Tenets of Platform-Based Design

Restricting the Space of Design Options

To which option will be best for a particular design, the designer will need to know:

- The ways in which each component option can be used.
- What it does (functionality) and simplified estimates of capability (performance),
- Cost and availability (which influences scheduling for time-to-market computations).
Flowchart of activities for platform-based design.

From a product perspective, product family success is achieved through ...

... a set of well-defined design modules and interfaces that can be easily customized to a variety of customer requirements.

From a process perspective, multi-project integration is achieved via

... a stable basis of processes and tools that support networks of cooperation with project partners and suppliers.
Design Efficiency and Correctness

Design is more efficient because ...

... an engineer working on abstraction level n can improve the quality of decision making by looking ahead to information at lower-level abstraction levels (n+1, n+2).

This reduces both the number of required iterations of development and, in particular, the need for large loop corrections.

Design is more correct because ...

... systems can only be assembled from components and connectors that have already been developed and work.

Economics improves because fewer design errors need to be corrected.
Design Platform Architectures

Two-part definition:

1. Decomposition of the overall functionality of a product into a set of defined functions and the component parts of the product that are going to provide those functions.

2. Specification of the interface between the components, in other words, how components are going to interact together in the product as a system.

Rules for Assembly of System Architectures

Small collection of modules and connectors.

Simple System Assembly
Elements of the System Platform Stack

Application Requirements
- Functional requirements
- Performance requirements
- Interface requirements
- Test requirements

Application Models
- Application behavior
- Application structure
- Application interfaces

Library of Design Options
- Component behavior
- Component structure
- Component interface

Application Space
- Family of Applications
- Design goals and constraints mapped to platform constraints
- Platform Interface defines the Explorable Design Space

Architecture Space
- Options for implementing the Physical System
- Platform Instances
Methodology

- When the gap between the application and architectural design spaces is too large, the task of finding good designs simply becomes too difficult.

Solution

- Organize design activities into a stack of abstractions.
Components of the platform will be ...

... **partially or completely pre-designed.**

**Component-Sharing Modularity**

Families of components with pre-defined characteristics are designed for ease of assembly.

Library of Components

- A
- B
- C

Design Alternative 1

Design Alternative 2
Component-Swapping Modularity

Families of components are designed with the goal of easily interfacing with a common interface (or bus).

System architecture is assembled around a common bus, and then reconfiguring the system by swapping two components of type A with those of type B.
Scale-to-fit Modularity

Components are defined in terms of a set of parameters that can be easily adjusted. Scale-to-fit modularity makes sense when we want to make a variety of products all off a single design.

Example 1. Scale-to-fit modularity for a family of swatches.
Example 2. Efficient generation of design alternatives for a family of seats.
Problem Statement: Analysis of heat conduction over an irregular domain.

In a two-dimensional setting Possions equation can be written:

$$\frac{\partial^2 \phi(x, y)}{\partial^2 x} + \frac{\partial^2 \phi(x, y)}{\partial^2 y} = f(x, y).$$  \hspace{1cm} (1)

over some domain $S$. Typically,

$$\phi(x, y) = f(x, y)$$  \hspace{1cm} (2)

along the boundary of $S$ (written $\partial S$).

Methods of Analysis

- Finite difference analysis,
- Finite element method.
Application 1: Platform Infrastructure

Application — Heat Conduction over an Irregular Domain

Engineering Analysis Instances

Finite difference analysis

Accuracy requirements

Finite difference analysis

Accuracy requirements

Geometric constraints

Mesh requirements

Boundary constraints

Finite difference modules

Finite element analysis

Selected finite elements

Finite element modules

Solver properties and capabilities

Solver requirements

Mesh requirements

Element requirements

Matrix analysis modules

Full equation solver

Banded equation solver

Skyline equation solver

Solver properties and capabilities

Implementation layer

C

Matlab

Java
Approach 1: Finite Difference Approximation

- **Basic Approach**
  Replace the continuous domain $S$ with a grid of discrete points, plus finite difference approximations to terms on the left-hand side of equation \[1\].
  Write family of difference equations in a matrix format.
  Solutions to equation \[1\] at the discrete grid points are computed through solutions to the matrix equations.

- **Pre-defined Components**
  Well-known finite difference strategies.
  Matrix equation solvers.

- **Key Decisions**
  Selection of the geometry and fidelity of the finite difference grid.
  Selection of an appropriate matrix equation solver.
  Selection of a domain for implementation.
Application 1: Engineering Analysis Platform

Approach 2: Finite Element Method

- **Basic Approach**
  Replace the continuous domain \( S \) with a mesh of finite elements. Finite elements are designed to provide a good approximation to the localized behavior of a larger continuous system.
  Solutions to equation 1 are computed at the nodal points of the individual elements.

- **Pre-defined Components**
  Well known triangular and quadrilateral finite elements.
  Matrix equation solvers.

- **Key Decisions**
  Selection of the geometry and fidelity of the finite element mesh.
  Selection of elements from the finite element library.
  Selection of an appropriate matrix equation solver.
  Selection of a domain for implementation.
Problem Statement

- Implement a stack of abstractions for the design, control, and operation of a pizza restaurant.

Issues

- How many layers of design abstraction will be needed?
- What will the library of reusable components look like at each layer of abstraction?
- What information will be passed between the layers?
- How will the platform framework benefit from having an ability to look down into design layers at lower levels of abstraction?
Application 2: Pizza Restaurant Design Platform

Proposed Platform Infrastructure

Factors Driving Design

- Architecture Requirements
- Occupancy requirements
- Safety requirements
- Control requirements
- Comfort requirements
- Required rate of pizza ordering.
- Required space for pizza assembly.
- Cooking control requirements
- Safety requirements
- Oven capacity requirements

Stack of Design Abstractions

- Architecture Design
- Restaurant Control Design
- Pizza Ordering System Design
- Pizza Assembly System Design
- Cooking Control Design
- Pizza Oven Design
- Implementation

Performance

- Maximum capacity
- Estimated hours of operation.
- Estimated capacity for ordering pizzas
- Maximum waiting time for pizza delivery
- Estimated hours of cooking operations
- Actual hours of operation
- Actual restaurant layout
- Actual capacity
- Actual oven design
- Actual cost

Legend

= Library of design options.
**Application 2: Pizza Restaurant Design Platform**

**Abstraction Layers.** The purposes of each layer are as follows:

1. **Architecture Design**
   - Functions: Allocation of spaces, their connectivity.
   - Design of the building exterior.

2. **Restaurant Control Design**
   - Functions: Hours of operation.

3. **Ordering System Design**
   - Functions: Take pizza orders. Collect payment from customers.

4. **Pizza Assembly System Design**
   - Functions: Roll pizza dough, add ingredients to dough.

5. **Cooking Control Design**
   - Functions: Set oven timer and temperature.

6. **Pizza Oven Design**
   - Functions: Bake pizza.
Platform Elements and Design Flows: Architecture Design

Typical Requirements

1. Target for maximum capacity of restaurant.
2. Spatial layout requirements (e.g., greeting area, dining area, ordering area, cooking and cleaning area, bathroom areas, storage area).

Library of Reusable Elements

1. Floorplans of similar restaurants.
2. Architectural rules-of-thumb for required space per customer.

Flowdown to Restaurant Control Design

1. Spatial layout constraints.
2. Estimates of customer demand versus time.

See class notes for details of remaining design layers....
Problem Statement

Modern buildings are ...

... advanced, self-contained and tightly controlled environments designed to provide a variety of services (e.g., vertical/horizontal transportation, sanitation, artificial lighting, fire protection, environmental conditioning, air quality, communication and security) to their occupants.

Modern buildings contain networks for ...

... the arrangement of spaces throughout the building, for the fixed circulatory systems (power, hvac, plumbing), for the dynamic circulatory systems (flows of energy through rooms; flows of material), and for wired and wireless communications.

Since many of these networks have their own independent behaviour, ...

... the overall problem for design automation is really a system of systems design problem.
Application 3: Building Automation Systems Design

System-of-Systems Approach to Building Automation Design

Source: Alberto Sangiovanni-Vincentelli, EE249, UC Berkeley, Fall Semester 2009.
Interaction of Discipline-Specific Concerns

**Architecture / Structural View**

External Factors
- Occupant functionality
- Performance metrics
- External environment

System Architecture
- Design, layout and connectivity of spaces....
- Building envelope / structural design

**Control View**

Occupancy demand.

Spatial constraints

Control System

Scheduling of thermal comfort, security, electrical and information services.

**Networked Embedded Systems View**

Thermal requirements
- Security requirements
- Electrical requirements
- Information requirements

Spatio–temporal constraints

Building Networks Design
- Selection, positioning and connectivity of networked embedded systems.

HVAC components
- Security components
- Computer components
- Electrical components

Feasibility of implementation
Traditional Top-Down Approach Automation Systems Design

Factors Driving Design

- Architectural constraints.
- Occupancy requirements.
- External loads (gravity, thermal).
- Control requirements.
- Comfort requirements.
- Control speed requirements.
- Sensor and actuator requirements.
- Layout requirements.

Design Flow

- Architecture Design
- Control Design
- Network Design
- Implementation

Flowdown of constraints

- Layout constraints.
- Sequence of operations.
- Actual implementation cost.

Traditional approaches to building automation design follow a top-down process with very little feedback.
Application 3: Building Automation Systems Design

Design Flow for Platform-Based Design

Factors Driving Design

Architectural requirements.
Occupancy requirements.
External loads (gravity, thermal, ...)

Ventilation requirements.
Energy generation requirements.

Sequence of operations.
Comfort requirements.

Control speed requirements.
Sensor and actuator requirements.

Layout requirements.

Design Flow

Architecture Design

Power Design

Control Design

Network Design

Implementation

Performance

Maximum ventilation.
Maximum power generation.
Cost estimates.

Minimum response time.
Control accuracy.

Maximum available bandwidth.
Maximum computational speed.
Maximum storage size.

Actual ventilation.
Actual power generation.
Actual network speed.
Actual layout constraints.
Actual installation cost.
Two Examples of Building-Integrated Energy Systems Design

The left-hand collage shows the Pearl River Complex currently under construction in Guangzhou, China.

The right-hand collage illustrates the use of wind and solar power mechanisms, and vacuum driven ventilation in the Chicago-bound Clean Technology Tower.
Research Question: To what extent can design patterns facilitate the development of an extensible framework for platform-based design?
Platforms for Engineering Biomedical Experiments

Source: Matt Mosteller’s Project Work in ENSE 622 (Sp. 2011) and ENSE 623 (Fall 2011)
Surface Reconstruction of a Bacterial Biofilm

Note: Highly variant nature of biofilm growth in commonly studied biological systems.
Platforms for Engineering Biomedical Experiments

Platform Architecture

Application Requirements
- Functional
- Performance
- Interface
- Test

Application Models
- Behavior
- Structure
- Interfaces

Application Space (Motivated by Biology)

Biomedical Application Instance
Family of Applications

Design Goals and Constraints Mapped to Platform Constraints

Platform Interface Defines the Explorable Design Space

Relevant Design Parameters

Options for Implementing the Physical System
Biomedical Device Architecture Instance

Architecture Space (Driven by Engineering)

- Component Behavior
- Component Structure
- Component Interface

Library of Design Options
Platforms for Engineering Biomedical Experiments

High- and Low-Level Abstractions for Design of Biomedical Systems

Key Points:

- Semi-Formal Models: High-level view of the complete system (efficiency).
- Formal Models: Detailed view of the actual system (accuracy).
Process Flow in a Typical Biomedical Experiment
Platforms for Engineering Biomedical Experiments

High-Level System Architecture for Biomedical-Device Performance

Key Points:

- System Inputs: environmental conditions; what is done during the experiment.
- Device System: (1) A way to connect with the biological system, (2) A way to control the biological experiment, (3) A way to integrate with the sensor network for detection.
- System Output: Families of experimental results.
SysML State Machines of Biomedical-Device Components
Platform Implementation for Engineering Experimental Biomedical-Device Systems

Model Using Markov Chain Paradigm

Model Using Traditional MBSE (e.g. SysML, UML)
Platforms for Engineering Biomedical Experiments

Experimental Setup

- Bacterial imaging is performed using confocal microscopy
- Microfluidic device is fabricated using soft lithography
- Sensors are charged-coupled devices with $128 \times 1$ pixel arrays.
Visual Representation of Biofilm Markov-Model Implementation
Platforms for Engineering Biomedical Experiments

Full-System schematic of the Integrated Microsystem

- PDMS
- Transparent Substrate & Gold Electrodes
- Wire-Board Connectors
- Parallel Experiments
- Microfluidic Channel
- Charge-Coupled Device Photodetectors
- PCB Platform
Spatio-Temporal Changes in Biofilm Optical Density

![3D Graph showing spatiotemporal changes in biofilm optical density with arrows indicating biofilm changes and steady biofilm growth.](image)
So What’s Next?

Platforms for Pediatric Cardiology?

UMCP has a mandate to work more closely with UM Hospital in Baltimore
So What’s Next?

What is a Congenital Heart Defect (CHD)?

- A child is born with an abnormally structured heart and/or large vessels.
- Hearts may have incomplete or missing parts, may be put together the wrong way, may have holes between chamber partitions or may have narrow or leaky valves or narrow vessels.

A Few Statistics

- Approximately one in fifty (one in a hundred) newborns have some form of congenital heart defect.
- Within the US, congenital heart defects affect 40,000 newborns each year.
- Congenital heart defects are responsible for one third of all birth defect-related deaths.
- 20 percent of children who make it through birth will not survive past their first birthday.

Source: http://childrensheartfoundation.org/research
So What’s Next?

The Circulatory System and Heart
Hyperplastic Left Heart Syndrome

**Normal heart**

- Small (hypoplastic) aorta
- Atrial septal defect (opening between the atria)

**Hypoplastic left heart syndrome**

- Patent (open) ductus arteriosus
- Small (hypoplastic) left ventriculus
Cardiac Magnetic Resonance Imaging (MRI)
So What’s Next?

Intervention with Surgery

Key Idea: Re-route pipes to improve efficiency of pumping and to prevent mixing of oxygenated and unoxygerenated blood.
Intervention with Stents

A zig-zag CP stent dilated with an 18 mm balloon-in-balloon.

State-of-the-Art Heart Modeling

Left-hand side: The epicardial surface of the 88 element finite element model of the right and left ventricles. The fibre orientations are shown as vectors on the surface.

Right-hand side: Principal strain vectors shown at end-diastole at midwall points of the finite element mesh. The transmurally directed vectors are compressive.

So What’s Next?

State-of-the-Art Heart Modeling and Visualization

Coronary arteries used for computation of blood flow, pressure and oxygen transport.

Calculated coronary arterial blood flow velocities are shown at the end of the four phases of the cardiac cycle. The darker shade indicates the coronary flow with a maximum value of 250mm/s.


