Product Line Design and Scheduling at Intel

Evan Rash and Karl Kempf
Decision Engineering Group
Intel Corporation
Agenda

1. Business Background
2. The Strategic Business Problem
3. Mathematical Formulation
4. Our New Solution
5. Our Custom Implementation
6. Growing Business Impact
Founded July 18th, 1968

Bob Noyce (co-inventor of the integrated circuit)
Gordon Moore (author of "Moore's Law")
Leading Edge Process Technology

- 65nm
- 45nm
- 32nm
- 22nm*

Lower Transistor Leakage

Higher Transistor Performance (Switching Speed)

* projected

"... more than 6 million 22nm transistors can fit in the period at the end of this sentence."
Leading Edge Product Technology

- 2 Billion transistors/core
- 4 cores/chip
Feature, Market, and Time Dynamics
Different Markets Need a Different Mix of Features

<table>
<thead>
<tr>
<th>Market</th>
<th>ASP</th>
<th>Vol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market1</td>
<td>$13</td>
<td>240,000</td>
</tr>
<tr>
<td>Market2</td>
<td>$15</td>
<td>300,000</td>
</tr>
<tr>
<td>Market3</td>
<td>$14</td>
<td>450,000</td>
</tr>
<tr>
<td>Market4</td>
<td>$12</td>
<td>880,000</td>
</tr>
<tr>
<td>Market5</td>
<td>$ 9</td>
<td>900,000</td>
</tr>
<tr>
<td>Marketing</td>
<td>ASP</td>
<td>Vol</td>
</tr>
</tbody>
</table>

Selling the product in the market brings in revenue
Different Markets Need a Different Mix of Features

Engineering and manufacturing incurs costs

<table>
<thead>
<tr>
<th>Feature1</th>
<th>Feature2</th>
<th>Feature3</th>
<th>Feature4</th>
<th>Feature5</th>
<th>Feature6</th>
<th>Eng &amp; Mfg</th>
</tr>
</thead>
<tbody>
<tr>
<td>$300,000</td>
<td>$400,000</td>
<td>$400,000</td>
<td>$250,000</td>
<td>$300,000</td>
<td>$200,000</td>
<td>Eng Cost</td>
</tr>
<tr>
<td>$1.50</td>
<td>$0.35</td>
<td>$1.25</td>
<td>$0.50</td>
<td>$0.50</td>
<td>$0.25</td>
<td>Mfg Cost / u</td>
</tr>
</tbody>
</table>

| Market1  | $13      | 240,000  |
| Market2  | $15      | 300,000  |
| Market3  | $14      | 450,000  |
| Market4  | $12      | 880,000  |
| Market5  | $9       | 900,000  |
| Marketing| ASP      | Vol      |
Different Markets Need a Different Mix of Features

<table>
<thead>
<tr>
<th>Markets</th>
<th>ASP</th>
<th>Vol</th>
<th>Feature1</th>
<th>Feature2</th>
<th>Feature3</th>
<th>Feature4</th>
<th>Feature5</th>
<th>Feature6</th>
<th>Eng &amp; Mfg</th>
<th>Eng Cost</th>
<th>Mfg Cost / u</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market1</td>
<td>$13</td>
<td>240,000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market2</td>
<td>$15</td>
<td>300,000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market3</td>
<td>$14</td>
<td>450,000</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market4</td>
<td>$12</td>
<td>880,000</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market5</td>
<td>$9</td>
<td>900,000</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Markets have unique feature requirements
Different Markets Have Different Timings

Markets are not all synchronized in time
Different Features Have Different Availabilities

Feature development must be synchronized with market windows
Constraints

- Feature sets in the products must meet (or exceed) the needs of the target markets
- Features must be engineered in time to be integrated into the products
- Products must be engineered and manufactured to hit the market timings
- The engineering budget is finite (leading to an emphasis on reuse)

Objective

- Maximize Profit  \((Max \ Revenue, \ Min \ Eng \ and \ Mfg \ Cost)\)
Business questions include (at least):

- Given an engineering budget, what set of products maximize revenue or profit?
- Given a revenue target, what set of products minimize cost, with what engineering budget?
- Given a number of Features to engineer, what is the profit maximizing order of development?
- Given a Feature ‘build vs. buy’ decision (cost, timing), which generates the most profit?

Difficult to solve with standard techniques due to many different constraints, competing objectives, and interrelated tradeoffs
Math

Define Problem & Formulate as Mathematical Programming

Show Complexity & Difficulties involved with Traditional techniques

Solution Methodology & Implementation
The Core Problem

Generate a Product Line  
*Strategic*
- Map products into markets
- Schedule product development

Generate Product Features  
*Tactical*
- Meet or exceed market requirements
- Schedule feature development

Optimize for Profitability  
*Strategic*
- Product line must optimize profitability
- Must consider engineering budgets
Generating the Product Line

Inputs

- Set of markets: \( \{1, \ldots, M\} \)
- Number of products: \( P \leq M \) - at most one product per market
- Time horizon: \( \{1, \ldots, T_0, \ldots, T\} \)

Decisions

- How many products to build: \( \beta_p \) - Binary
- When to introduce products: \( z_p \in \{T_0, \ldots, T\} \) - Integer
- Which markets to sell products into? \( \alpha_{pmt} \) - Binary
Inputs
- Set of features \( \{1, \ldots, F\} \)
- Market Requirements \( D_{mf} \)

Decisions
- **Product Features** (Units of Feature \( f \) in Product \( p \)) \( x_{pf} \) \( \text{Integer} \)
- **Feature Availability** \( y_f \in \{T_0, \ldots, T\} \) \( \text{Integer} \)
Optimize for Profitability

Inputs
Market Volumes and Prices

Feature Engineering Cost (with Reuse)

Product Engineering Cost

Feature Mfg. Cost

Expressions
Revenue

Engineering Cost

$$v_{mt}, p_{mt}$$

$$R_f(t)$$

$$A$$

$$c_f$$

$$\sum_{p=0}^{P} \sum_{m=0}^{M} \sum_{t=0}^{T} \alpha_{pmt} v_{mt} \left( p_{mt} - \sum_{f=0}^{F} c_f x_{pf} \right)$$

$$A \sum_{p=0}^{P} \beta_p + \sum_{f=0}^{F} \sum_{t=0}^{T} R_f(t)$$
Reuse Function

Engineering features presents **reuse** opportunities

Developing **Feature 3** may cause developing **Feature 4** to be cheaper/faster

The **Reuse Function** defines these reuse synergies

Typically dynamic and complex
Reuse Function Example

A hypothetical Reuse Function where developing one feature in a group causes subsequent feature development to be 50% cheaper.

<table>
<thead>
<tr>
<th>Feature $f$</th>
<th>Group $G(f)$</th>
<th>$R_f(t)$ when $t - 1 = y_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>.5 if $y_2 &gt; y_3$, else 1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>.5 if $y_3 &gt; y_2$, else 1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>.5 if $y_4 &gt; y_5$, else 1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>.5 if $y_5 &gt; y_4$, else 1</td>
</tr>
</tbody>
</table>

$$R_f(t) = \begin{cases} 1 & \text{if } y_f = \min_{g \in G(f)} y_f \\ .5 & \text{if } \exists g \in G(f) | y_g < y_f \end{cases}$$
Full Formulation

**Objective: Maximize Profit**

\[
\max \sum_{p=0}^{P} \sum_{m=0}^{M} \sum_{t=0}^{T} \alpha_{pmt} V_{mt} \left( P_{mt} - \sum_{f=0}^{F} C_{f} x_{pf} \right) - \sum_{f=0}^{F} \sum_{t=0}^{T} R_{f}(t) - A \sum_{p=0}^{P} \beta_{p}
\]

**Subject to:**

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{p=0}^{P} \alpha_{pmt} \leq 1 ) ( \forall m, t )</td>
<td>One Product per Market</td>
</tr>
<tr>
<td>( D_{mf} \alpha_{pmt} \leq x_{pf} ) ( \forall p, m, t )</td>
<td>Market Satisfaction Constraint</td>
</tr>
<tr>
<td>( z_{p} \geq \max{f: \beta_{p} &gt; 0</td>
<td>y_{f}} )</td>
</tr>
<tr>
<td>( \alpha_{pmt} = 0 ) ( \forall p, m, t &lt; z_{p} )</td>
<td>Market Coverage Availability Constraint</td>
</tr>
<tr>
<td>( MT \beta_{p} \geq \sum_{m=0}^{M} \sum_{t=0}^{T} \alpha_{pmt} )</td>
<td>Product Selling Requirement</td>
</tr>
<tr>
<td>( \sum_{f=0}^{F} R_{f}(t) \leq S_{t} )</td>
<td>Resource Constraint</td>
</tr>
<tr>
<td>( \beta_{p} \in {0,1} )</td>
<td>Binary Constraint</td>
</tr>
<tr>
<td>( \alpha_{pmt} \in {0,1} )</td>
<td>Binary Constraint</td>
</tr>
<tr>
<td>( x_{pf} \in {0, \ldots, \max D_{mf} } )</td>
<td>Integral Units of Features Constraint</td>
</tr>
<tr>
<td>( y_{f} \in {T_{0}, \ldots, T + \Pi} )</td>
<td>Scheduling Window Constraint</td>
</tr>
<tr>
<td>( z_{p} \in {T_{0}, \ldots, T + \Pi} )</td>
<td>Scheduling Window Constraint</td>
</tr>
</tbody>
</table>
Why Is This a Hard Problem?

**Non-linearity**
- Reuse Function
- Objective Function & Constraints

**Integral & Binary Decisions**
- Scheduling
- Mapping

**Combinatorics & Problem Size**

*Difficult to solve by traditional techniques!*

*Linear/Mixed-Integer Programming*

*Constraint Programming*
Our Solution

• Integrate diverse OR techniques
  – Resource-Constrained Job Scheduling
  – Optimal Set Covering
  – Portfolio Optimization
  – Dynamic Programming

• Decompose Problem into Multiple Stages
  – Outer “strategic” Genetic Algorithm
  – Inner “tactical” Heuristics and MIPs
  – Financial Optimization through Genetic Algorithm Fitness
Decomposition – Product Line Design

Outer Genetic Algorithm

1. Product Release Schedule
2. Product to Market Mappings

Inner Set Covering/Heuristics

3. Product Features
   Ext: Feature Substitutions
4. Feature Schedules
5. Feature Reuse
6. Resource Constraints
7. Financial Objectives
Decomposition – Generate Product Features

Outer Genetic Algorithm

1. Product Release Schedule
2. Product to Market Mappings

Inner MIPs and Heuristics

3. Product Features
   - Ext: Feature Substitutions
4. Feature Schedules
5. Feature Reuse

6. Resource Constraints
7. Financial Objectives

Decomposition – Financial Optimization

**Outer Genetic Algorithm**

*Mutation & Crossover*

1. Product Release Schedule

2. Product to Market Mappings

*Selection*

6. Resource Constraints

7. Financial Objectives

**Inner Set Covering/Heuristics**

*Per Product*

3. Product Features
   - *Ext: Feature Substitutions*

4. Feature Schedules

5. Feature Reuse

*Per Product*

© Intel Corp 2011

CPMS 2011 Daniel H. Wagner Prize Competition
Outer “Strategic” Algorithm

1. **Outer: Creating Product Schedules**

   Generate a random chronologically sorted product schedule, with some products “turned off”.

   Use crossover to “zip” different schedules together and mutations to randomly permute schedule by pushing products out and pulling products in.

2. **Outer: Creating Market to Product Mappings**

   For each market randomly cover or skip the market. If covered, select a random product from the list generated in 1.
**Inner “Tactical” Algorithm**

3. **Inner: Determine Product Features (MIPs)**
   - Cover market requirements with minimum manufacturing cost
   - Cover market requirements with minimum engineering cost
   
   Randomly alternate and allow the evolutionary process to pick the best

4. **Inner: Deduce Feature Schedules**
   
   Back out the feature engineering schedule based on when the features need to be available for the product’s availability (1)

5. **Inner: Evaluate Reuse**
   
   Evaluate the reuse of the feature schedule from (4)
6. **Evaluate Resource Constraint**

   Evaluate the engineering resources for the entire roadmap

   Model engineering resource constraints as soft constraints

   Use a Lagrangian penalty approach similar to the concept of an “overtime” cost of exceeding the available engineering resource supply

7. **Evaluate NPV & Fitness**

   Evaluate the fitness of the product line by determining its NPV and subtracting out any resource overage penalties
“Pinning” Parts of the Solution

• Planning involves many strategic aspects
  – Not always possible to solve with a “clean slate”
• Solver must be able to “pin” portions of the solution in place and solve using remaining degrees of freedom
• Examples
  – Locking products onto the roadmap
  – Locking feature availability schedules
  – Forcing entry into particular markets
Implementation

- Custom Implementation (C# .NET)
  - Required Custom Mutation/Crossover and Solution Flow
- Inner sub-problem solved via modular heuristics plugged into larger GA
  - Most Heuristics: C#
  - Feature Substitution: OPL CPLEX
<table>
<thead>
<tr>
<th>BEFORE</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Many spreadsheets with local databases</td>
<td>1) One tool with global database (HW and SW)</td>
</tr>
<tr>
<td>2) Local view by product, sometimes by division</td>
<td>2) Holistic view across divisions and products</td>
</tr>
<tr>
<td>3) Few what-ifs</td>
<td>3) Many what-ifs</td>
</tr>
</tbody>
</table>
## The Business Process

<table>
<thead>
<tr>
<th>BEFORE</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Many spreadsheets with local databases</td>
<td>1) One tool with global database (HW and SW)</td>
</tr>
<tr>
<td>2) Local view by product, sometimes by division</td>
<td>2) Holistic view across divisions and products</td>
</tr>
<tr>
<td>3) Few what-ifs</td>
<td>3) Many what-ifs</td>
</tr>
<tr>
<td>4) Difficult decision making between finance, planning, and engineering (design and mfg)</td>
<td>4) Collaborative decision making between all of the product functions</td>
</tr>
<tr>
<td>5) No global optimization and little (if any) local optimization</td>
<td>5) Global profit optimization</td>
</tr>
<tr>
<td>6) Little reuse between divisions and within divisions</td>
<td>6) Increasing reuse across divs and products (few%/mo)</td>
</tr>
</tbody>
</table>
User Data and Feedback

USERS BY MONTH

07-10 08-10 09-10 10-10 11-10 12-10 01-11 02-11 03-11 04-11 05-11 06-11 07-11 08-11 09-11 10-11
User Data and Feedback

**USERS BY MONTH**

- **07-10**: 0
- **08-10**: 20
- **09-10**: 40
- **10-10**: 60
- **11-10**: 80
- **12-10**: 100
- **01-11**: 120
- **02-11**: 140
- **03-11**: 160
- **04-11**: 180
- **05-11**: 200
- **06-11**: 220
- **07-11**: 240
- **08-11**: 260
- **09-11**: 280
- **10-11**: 300

**“No idea how we can optimize market coverage without a tool like this.”**

**“We are finally working in a transparent system instead of spreadsheets on random shared drives.”**

**“Useful as an acumen tool as well as learning about where synergies exist for our products.”**

**JOB TITLE** | **# USERS** | **HRS USED**
--- | --- | ---
STRATEGIC PLANNERS | 16 | 996
PROJECT/PROGRAM MANAGER | 39 | 476
PRODUCT DESIGN ENGINEER | 23 | 394
FINANCIAL ANALYST | 39 | 342
PRODUCT MARKETING ENGINEER | 5 | 63
OPERATION MANAGER | 3 | 21
PRODUCT SOFTWARE ENGINEER | 8 | 4

**TOTAL** | **133** | **2296**

**PRODUCT** | **# OF USERS**
--- | ---
Div-1 | 58
Div-2 | 25
Div-3 | 12
Div-4 | 11
Div-5 | 9
Div-6 | 4
Div-7 | 3
Div-8 | 3
Div-9 | 2
Div-Admin | 3
Misc | 3

**TOTAL** | **133**
Conclusion

• This is a complex problem considering market, feature, and product time dynamics
• Extremely difficult to solve with traditional techniques
• Developed and implemented a custom solution to the problem
• The system currently has users across divisions and job roles
• We believe the system (over time) will become crucial to Intel’s continuing success
IT'S WHAT WE MAKE POSSIBLE.
Extensions

- Feature Substitution
  - Feature A or Feature B can be interchanged
- Time to Market Penalties
  - Late products suffer in the marketplace
- Minimum vs. Target Market Requirements
  - Feature A is a must-have, Feature B is a value-add
- Build vs. Buy decisions
  - Develop in house or license?
- NPV Optimization