EXECUTABLE BEHAVIORAL MODELING OF SYSTEM-AND SOFTWARE-ARCHITECTURE SPECIFICATIONS TO INFORM RESOURCING DECISIONS

Monica Farah-Stapleton, PhD
Mikhail Auguston, PhD
Kristin Giammarco, PhD
Ray Madachy, PhD
• Statement of Problem
• Research Objectives
• Summary of Results and Findings
• ThreeMetrics Methodology
• Research
  – Methodology Description
  – Examples
• Closing Thoughts
• Discussion
• Information Technology (IT) systems are large, complicated, and represent a significant investment in time and resources
  – Operational and financial impacts are often assessed after the fact
  – Resourcing decisions and precise architectural descriptions of the system and environment are often minimally related

• Precise modeling of architectures highlights design details and offers an early cost estimate for the system design
  – Assists in assessing architectural design decisions and their impacts prior to, during, and after implementation and deployment
  – Descriptions of interactions are related to function point transactional and data functions, and offer a way to estimate effort and cost of design decisions early in the process
  – Provides foundation for supporting unadjusted function point (UFP) count with automated tools
This research developed a methodology to extract unadjusted function point (UFP) counts from executable architectural behavioral models, for use in cost estimation models such as COCOMO II, in order to inform effort estimates early in the life cycle.
Summary of Results and Findings

- The ThreeMetrics methodology:
  - Is able to extract an unadjusted function point (UFP) count from Monterey Phoenix’s (MP) executable architecture models for use in software cost estimation
  - Leverages precise behavioral modeling using MP and the MP Analyzer on Firebird to assess architecture design decisions and their impacts
  - Relates architecture modeling to resourcing through UFP counts
  - Uses event traces to inform integration test estimates and decision making

- The COCOMO II tool is used to input the UFP count to determine cost estimates

- Each step of the ThreeMetrics methodology provides meaningful information to stakeholders
Determine the Type of Count and Boundary:
Development Project, Enhancement Project, Application

Terminology:
- External Inputs (EI): Data that is entering the system
- External Outputs (EO) and External Inquiries (EQ): Data that is leaving the system
- Internal Logical Files (ILF): Data that is processed and stored within the system
- External Interface Files (EIF): Data that is maintained outside the system but is necessary to satisfy a particular process requirement

Function Point Analysis Practice
1. Count Data and Transactional Function Types
2. Determine UFP Count
3. Determine the Value Adjustment Factor
4. Calculate final Adjusted FP Count

Source: International Function Point User Group (IFPUG)
ThreeMetrics Methodology

Step 1: Determine Stakeholder Questions To Be Answered and Gather Existing Documentation

Step 2: Identify Scope and Application Boundary

Step 3: Develop MP Model

Step 4: Extract Data Functions Count From MP Model

Step 5: Extract Transactional Functions Count From MP Model

Step 6: Extract Integration Test Cases and Views From MP Model

Step 7: Determine Unadjusted FP Count

Step 8: Calculate Effort Estimate

Step 9: Finalize Analysis and Provide Results to Stakeholders

ThreeMetrics contribution
Step 1: Determine Stakeholder Questions To Be Answered and Gather Existing Documentation

- Identify questions of stakeholders (e.g., customers, users, testers, engineers, designers, cost analysts)
- Gather documentation and subject matter expert input
- Determine if the application is new or an enhancement to an existing application
Step 2: Identify Scope and Application Boundary
ThreeMetrics Box and Arrow View

- Identify boundaries of application to be counted
  - Interactions mark the boundaries
  - If there is an interaction, there is a dependency in one or both directions
  - Capture interactions in separate parts of MP model
- Identify data functions (ILF, EIF)
- Identify transactional functions (EI, EQ, EO)
- Identify Internal Abstracted Application (IAA)
- Environment
  - Everything but the application under assessment
  - Interactions with the application
Step 3: Develop MP Model

- MP operates in the event space
  - Behaviors of processes and data are uniquely distinguished events

- Describe ROOTs
  - Actors

- Represent data functions and data element types as actors

- Describe interactions between actors using COORDINATE and SHARE ALL composition operations
  - Transactional functions
  - Data functions
Step 4: Extract Data Functions Count From MP Model

- Inspect MP model for number of interactions for data functions (ILF, EIF)
- Count number of SHARE ALLs, for each data function
- Count number of COORDINATEs and number of ADDs if detailed source information is available, for each data function
Step 5: Extract Transactional Functions Count
From MP Model

- Inspect MP model for interactions between processes (EI, EO, EQ)
- Count number of COORDINATEs for each transactional function
- Count number of COORDINATEs and number of ADDs if detailed source information is available, for each transactional function
Event traces (i.e., use cases) are automatically generated by MP Analyzer on Firebird.

MP Analyzer provides exhaustive set of use cases for given scope.

According to Brooks, integration testing comprises up to ¼ of effort.

Inspect event traces and identify which ones can be used as a blueprint for integration test case design.
Step 7: Determine UFP Count

- Use IFPUG tables
  - Functional complexity
  - Functional size values are associated with functional complexity
- Complexity in MP model is determined by number of ADDs counted in COORDINATE composition operation
- For minimal source information, initially assume Average functional complexity and size to calculate the total UFP count
- Total UFP count is summation of UFP count for each data function and UFP for each transactional function
Step 8: Determine Effort Estimate

- Manually insert UFP into COCOMO II calculator tool or import .mp file
- Select
  - Sizing method (e.g., function points)
  - Input method (e.g., direct)
  - Language (e.g., Java)
  - Insert cost per person-month (e.g., 20,000 dollars)
  - Nominal options
- Interpret results for effort, schedule, and cost
Step 9: Finalize Analysis and Present Results to Stakeholders

- Provide analysis results and views of information in formats for each group of stakeholders, including
  - Box and arrow diagrams
  - Sequence diagrams
  - High-level pseudocode
  - Cost/schedule/effort values
The ThreeMetrics methodology was applied to 3 FP counting examples:

1. Spell Checker
   - Limited source information. Use Average functional complexity and size values for transactional and data functions. Inspect model for COORDINATE and SHARE ALL.

2. Course Marks
   - Limited source information. Use functional complexity and size values provided in source information, for transactional and data functions. Inspect model for COORDINATE and SHARE ALL.

3. It’s Tee Time
   - Detailed source information available, explore 4 courses of action (COAs) to determine UFP count.
     - COA 1: Inspect model for COORDINATE and SHARE ALL. Assume Average functional complexity and size values.
     - COA 2: Inspect model for EI, EO, and EQ descriptive terms with COORDINATE and SHARE ALL. Assume Average functional complexity and size values.
     - COA 3: Inspect model for COORDINATE and ADD to determine functional complexity and size for transactional functions. Inspect model for SHARE ALL assuming Average functional complexity and size for data functions.
     - COA 4: Inspect model for COORDINATE and ADD for both transactional and data function functional complexity and size.
Step 1: Determine Stakeholder Questions To Be Answered and Gather Existing Documentation

**Transaction Function: EQ (View or Display Retrieval of Data)**
- Click Button from Main Screen, navigate to Golf Courses list;
- Exit button: Navigation

<table>
<thead>
<tr>
<th>EQ – View/Display State Drop Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ Click on state arrow</td>
</tr>
<tr>
<td>➤ State list display returned</td>
</tr>
<tr>
<td>➤ Stop</td>
</tr>
<tr>
<td>➤ 1 FTR (Golf Courses ILF), 2 DET (Arrow Click, State field)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EQ – View/Display City Drop Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ State data entered</td>
</tr>
<tr>
<td>➤ Click on City arrow</td>
</tr>
<tr>
<td>➤ City list display returned</td>
</tr>
<tr>
<td>➤ Stop</td>
</tr>
<tr>
<td>➤ 1 FTR (Golf Courses ILF), 3 DET (State data, Arrow Click, City field)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EQ – View/Display Golf Courses List</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ Enter State and City</td>
</tr>
<tr>
<td>➤ Click Display button</td>
</tr>
<tr>
<td>➤ Name displayed back, don’t count State, City</td>
</tr>
<tr>
<td>➤ Stop</td>
</tr>
<tr>
<td>➤ 1 FTR (Golf Course ILF), 4 DET(State, City, Name, Click Display)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EP</th>
<th>Description</th>
<th>ILF/EIF</th>
<th>FTR/DET</th>
<th>Complex</th>
<th>UFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>State Drop Down</td>
<td>Golf Courses (I)</td>
<td>(1,2)</td>
<td>Low</td>
<td>3</td>
</tr>
</tbody>
</table>
Step 2: Identify Scope and Application Boundary
ThreeMetrics Box and Arrow View

Source: Derived from the It’s TeeTime FP counting example, protected by copyright, provided by Q/P Management Group Inc. for this research
Step 3: Develop MP Model

EQ State Drop Down Transactional Function Example

Line 01 Schema Name for EQ

01 SCHEMA It_Is_Tee_Time_EQ

02 ROOT User_GCL: **Inquire_on_state_data** something_else;

03 Inquire_on_state_data: Click_state_arrow_dropdown Receive_state_list_display;

04 ROOT Golfcourses_ILF: **Get_result** anything_else;

05 Get_result: Receive_state_arrow_prompt Send_state_list_display;

06 COORDINATE $x: Inquire_on_state_data FROM User_GCL,

07 $y: Get_result FROM Golfcourses_ILF

08 DO

09 COORDINATE $xx: Click_state_arrow_dropdown FROM $x,

10 $yy: Receive_state_arrow_prompt FROM $y,

11 $x11: Receive_state_list_display FROM $x,

12 $y11: Send_state_list_display FROM $y

13 DO

14 ADD $xx PRECEDES $yy;

15 ADD $y11 PRECEDES $x11;

16 OD;

17 OD;

MP Calculation

- 1 FTR and 1 nested COORDINATE (COORDINATE & 2 ADDs) correspond to 1 FTR and 2 DETs and a functional complexity weighting of 3
- EQ State Drop Down = (1 COORDINATE) * 3 UFP/COORDINATE = 3 UFPs
Step 3: Develop MP Model, con’t
Golf Courses ILF Data Function Example

Data function behavior is represented using either
• SHARE ALL, if minimal ILF or EIF source information is available
• COORDINATE, if more detailed ILF or EIF source information is available

**SHARE ALL**

01 SCHEMA Data_Function_Example_SHARE_ALL
02 ROOT TT: (* ( writing | reading) *);
03 ROOT GC_ILF: (+writing+) ( * sending *);
04 TT, GC_ILF SHARE ALL writing;

**COORDINATE**

01 SCHEMA Data_Function_ExampleCOORDINATE
02 ROOT TT: (* ( start_writing | reading) *);
03 ROOT GC_ILF: (+finish_writing+) ( * sending *);
04 COORDINATE $a: start_writing FROM TT,
05 $b: finish_writing FROM GC_ILF
06 DO ADD $a PRECEDES $b; OD;
Step 4: Extract Data Functions Count From MP Model

- Inspect MP model for SHARE ALL or COORDINATE

- MP Schema, Tee Time COA 3
  - Count number of data function SHARE ALLs (4)

- MP Schema, Tee Time COA 4
  - Count number of data function COORDINATEs (4)
  - Count number of ADDs in each COORDINATE to determine complexity
Step 5: Extract Transactional Functions Count From MP Model

- Inspect MP model for COORDINATE

- MP Schema, Tee Time COA 4
  - Count number of transactional function COORDINATEs (20)
  - Count number of ADDs in each COORDINATE to determine complexity
Step 6: Extract Integration Test Cases and Views From MP Model

• Extract Views
  – Expresses dependencies between parts of the system
  – Highlight modifications to the system
  – Event traces (use cases) generated by Firebird
  – Inspect and use to inform test case generation

• Extract Integration Test Cases
  – Integration test case creation + execution = Estimate for integration testing
  – Brooks “1/3 planning, 1/6 coding, 1/4 component test and early system test, 1/4 system test, all components in hand.”
  – .25 x Total effort = Estimate for integration testing
  – Wolff suggests 6 integration test cases can be executed per day for a large application (e.g., large commerce system)
• Integration and Test costs are part of COCOMO II Construction phase effort
  – Construction phase is allocated 6.4 months of schedule
  – 25% of that time is 1.6 months
  – Assume 5 days per week and 8 hours per day for each staff person
• Assume execution of 6 test cases per day
  – 192 test cases can be executed in the allocated time
  – Doesn’t include creation of test cases
• 864 event traces generated in the MP model for COA 1
  – Require over 144 days to execute all testing
  – Some event traces are significantly less complicated than others, so effort will vary
• Informs next steps for technical and programmatic decision making
  – Revisit the model
  – Determine if there are any additional resources (e.g. schedule relief, staffing)
  – Identify a subset of test cases for creation and execution
    • 144 days is unsupportable given schedule allocation
    • 192 test cases do not account for creation and analysis, only execution
UFP Calculation: FP Counting

- 1 FTR and 2 DETs identified from the behavior of EQ State_Drop_Down
- 0-1 FTRs and 1-5 DETs correspond to a Low functional complexity rating
- A Low functional complexity rating corresponds to 3 UFPs

UFP Calculation: Extracted From MP

- 1 COORDINATE interaction associated with State Drop Down EQ behaviors
- State Drop down EQ COORDINATE contains a nested COORDINATE (2 ADDs)
- The 2 ADDs relate to 2 DETs
- ROOT Golfcourses_ILF relates to 1 FTR
- 0 -1 FTRs and 1-5 DETs correspond to a Low functional complexity rating
- A Low functional complexity rating corresponds to 3 UFPs
- EQ State Drop Down is equal to 1 COORDINATE with a weight of 3 or 3 UFPs
Step 7: Determine Unadjusted FP Count, con’t

This table contains the results of the UFP count extracted from the It’s Tee Time MP model, using the ThreeMetrics:

- Total UFP count is summation of UFP count for each data function and UFP for each transactional function.
- UFP for Tee Time
  - 62 UFPs for transactional functions (EI, EO, EQ) (COAs 3 and 4)
  - 26 UFPs for data functions (ILF, EIF)
  - Total UFP count = 62 + 26 = 88 UFPs (COA 4)
<table>
<thead>
<tr>
<th>Example Name</th>
<th>Representation of Data Function Types in MP</th>
<th>Total # Data Function Type UFP</th>
<th>Representation of Transactional Function Type in MP</th>
<th>Total # Transactional Function Type UFP</th>
<th>ThreeMetrics Total UFP Count</th>
<th>Source Info Total Key UFP Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spell Checker</td>
<td>SHARE ALL</td>
<td>24</td>
<td>COORDINATE</td>
<td>31</td>
<td>55</td>
<td>58</td>
</tr>
<tr>
<td>Course Marks</td>
<td>SHARE ALL</td>
<td>15</td>
<td>COORDINATE</td>
<td>24</td>
<td>39</td>
<td>35 or 34</td>
</tr>
<tr>
<td>It’s Tee Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- COA 1</td>
<td>SHARE ALL</td>
<td>34</td>
<td>COORDINATE</td>
<td>86</td>
<td>120</td>
<td>88</td>
</tr>
<tr>
<td>- COA 2</td>
<td>SHARE ALL with ILF and EIF keywords</td>
<td>37</td>
<td>COORDINATE with EI, EO, EQ keywords</td>
<td>82</td>
<td>119</td>
<td>88</td>
</tr>
<tr>
<td>- COA 3</td>
<td>SHARE ALL with ILF and EIF keywords</td>
<td>37</td>
<td>Nested COORDINATE and ADD, with EI, EO, EQ keywords</td>
<td>62</td>
<td>99</td>
<td>88</td>
</tr>
<tr>
<td>- COA 4</td>
<td>COORDINATE with ILF and EIF keywords</td>
<td>26</td>
<td>Nested COORDINATE and ADD, with EI, EO, EQ keywords</td>
<td>62</td>
<td>88</td>
<td>88</td>
</tr>
</tbody>
</table>
Step 8: Determine Effort Estimate
Tee Time COA 4 COCOMO II Nominal Effort Options

Cost per Person-Month:
$20,000 selected as average labor rate
Reviewed GSA rates for several projects

MP/COCOMO tool created by Dr. Ray Madachy, NPS
rjmadach@nps.edu
• Sizing Method: Function Points
• UFP: 88
• Nominal options
• Software Development
  – Elaboration and Construction Phases
  – Effort: 16.0 Person-months
  – Schedule: 9.2 Months
  – Cost: $319,750
• Total Equivalent Size: 4664 SLOC
  – SLOC = (# UFP) x (language SLOC/UFP)
  – Java language conversion ratio for UFP to SLOC is 53
  – 4664 SLOC = (88 UFP) x (53 conversion ratio)
Step 9: Finalize Analysis and Present Results to Stakeholders

- Each step of the ThreeMetrics methodology provides meaningful information to stakeholders
- High-level pseudocode
- Resourcing requirements presented with each instance of architecture model
- Views extracted from architecture model
  - Use cases (event traces)
  - Sequence diagrams
Closing Thoughts

• The ThreeMetrics methodology:
  – Does relate function point counting, COCOMO II cost estimates, and executable behavioral modeling of system and software architecture specifications.
  – Leveraging the MP architecture model, provides a way of establishing internal and external boundaries for function point counting.
  – Successfully unifies the two distinct function point counting concepts of data function types and transactional function types.

• The use of the MP language and framework significantly simplified otherwise complex relationships -- the ability to execute the model using MP Analyzer on Firebird, inspect it, and debug it, provided confidence in the results of the model.

• Future work:
  – Determine criteria for selection of integration test cases from set of available MP event traces, leveraging Jackson’s small scope hypothesis
  – Determine selection of COCOMO II formula coefficients based on MP metrics
  – Consider how to use all the information in an MP model, particularly the event grammar (“pseudo code”), to inform cost estimates
Discussion
Back Up
Peer Reviewed Conference Papers and Presentations


Invited Presentations


Peer Reviewed Technical Posters


Invited Article

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COA</td>
<td>Course of Action</td>
</tr>
<tr>
<td>COCOMO II</td>
<td>Constructive Cost Model II</td>
</tr>
<tr>
<td>DET</td>
<td>Data Element Type</td>
</tr>
<tr>
<td>DOD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DODAF</td>
<td>DoD Architecture Framework</td>
</tr>
<tr>
<td>EI</td>
<td>External Input</td>
</tr>
<tr>
<td>EIF</td>
<td>External Interface File</td>
</tr>
<tr>
<td>EO</td>
<td>External Output</td>
</tr>
<tr>
<td>EQ</td>
<td>External Inquiry</td>
</tr>
<tr>
<td>FP</td>
<td>Function Point</td>
</tr>
<tr>
<td>FPA</td>
<td>Function Point Analysis</td>
</tr>
<tr>
<td>FPC</td>
<td>Function Point Counting</td>
</tr>
<tr>
<td>FTR</td>
<td>File Type Referenced</td>
</tr>
<tr>
<td>IAA</td>
<td>Internal Abstracted Application</td>
</tr>
<tr>
<td>IFPUG</td>
<td>International Function Point User Group</td>
</tr>
<tr>
<td>ILF</td>
<td>Internal Logical File</td>
</tr>
<tr>
<td>RET</td>
<td>Record Element Type</td>
</tr>
</tbody>
</table>
## FPA Calculation
### External Inquiry (EQ)

<table>
<thead>
<tr>
<th>EP</th>
<th>Description</th>
<th>ILF/EIF</th>
<th>FTR/DET</th>
<th>Complexity</th>
<th>UFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>State Drop Down</td>
<td>Golf Courses (I)</td>
<td>(1,2)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>City Drop Down</td>
<td>Golf Courses (I)</td>
<td>(1,3)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Golf Course List</td>
<td>Golf Courses (I)</td>
<td>(1,4)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Golf Course Detail</td>
<td>Golf Courses (I)</td>
<td>(1,12)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Scoreboard Display</td>
<td>Scoreboard (I)</td>
<td>(1,6)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Maintain Golf Course Display (by ID)</td>
<td>Golf Courses (I)</td>
<td>(1,13)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Tee Times Reservation Display</td>
<td>Tee Times (I)</td>
<td>(1,11)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Tee Times Shopping Display</td>
<td>Merchandise (E)</td>
<td>(1,3)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>Product View Picture</td>
<td>Merchandise (E)</td>
<td>(1,3)</td>
<td>Low</td>
<td>3</td>
</tr>
</tbody>
</table>

### Source
## FPA Calculation

### External Input (EI)

<table>
<thead>
<tr>
<th>EP</th>
<th>Description</th>
<th>ILF/EIF</th>
<th>FTR/DET</th>
<th>Complexity</th>
<th>UFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>Scoreboard (ADD)</td>
<td>Scoreboard (I)</td>
<td>(1,7)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Scoreboard (CHANGE)</td>
<td>Scoreboard (I)</td>
<td>(1,7)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Scoreboard (DELETE)</td>
<td>Scoreboard (I)</td>
<td>(1,3)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Maintain Golf Course (ADD)</td>
<td>Golf Courses (I)</td>
<td>(1,13)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Maintain Golf Course (CHANGE)</td>
<td>Golf Courses (I)</td>
<td>(1,13)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Maintain Golf Course (DELETE)</td>
<td>Golf Courses (I)</td>
<td>(2,3)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Tee Times Reservations (ADD)</td>
<td>Tee Times (I)</td>
<td>(1,12)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Tee Times Reservations (CHANGE)</td>
<td>Tee Times (I)</td>
<td>(1,12)</td>
<td>Low</td>
<td>3</td>
</tr>
<tr>
<td>EI</td>
<td>Tee Times Reservations (DELETE)</td>
<td>Tee Times (I)</td>
<td>(1,6)</td>
<td>Low</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EP</th>
<th>Description</th>
<th>ILF/EIF</th>
<th>FTR/DET</th>
<th>Complexity</th>
<th>UFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EO</td>
<td>Shopping Display (Calculation)</td>
<td>Merchandise</td>
<td>(1,7)</td>
<td>Low</td>
<td>4</td>
</tr>
<tr>
<td>EO</td>
<td>Buy – Send to Purchasing (Calculation)</td>
<td>Merchandise</td>
<td>(1,15)</td>
<td>Low</td>
<td>4</td>
</tr>
</tbody>
</table>

**External Outputs:**

<table>
<thead>
<tr>
<th>Functional Complexity Rating</th>
<th>Function Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>5</td>
</tr>
<tr>
<td>High</td>
<td>7</td>
</tr>
</tbody>
</table>

### Functional Complexity of Each Data Function

<table>
<thead>
<tr>
<th>DET Range</th>
<th>1 to 19 DETs</th>
<th>20 to 50 DETs</th>
<th>51 or More DETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 RET</td>
<td>Low</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>2 to 5 RETs</td>
<td>Low</td>
<td>Average</td>
<td>High</td>
</tr>
<tr>
<td>6 or More RETs</td>
<td>Average</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

### ILF Translation Table: Use the following table to assign a functional size to each ILF.

<table>
<thead>
<tr>
<th>Functional Complexity Rating</th>
<th>Function Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>7</td>
</tr>
<tr>
<td>Average</td>
<td>10</td>
</tr>
<tr>
<td>High</td>
<td>15</td>
</tr>
</tbody>
</table>

### EIF Translation Table: Use the following table to assign a functional size to each EIF.

<table>
<thead>
<tr>
<th>Functional Complexity Rating</th>
<th>Function Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>5</td>
</tr>
<tr>
<td>Average</td>
<td>7</td>
</tr>
<tr>
<td>High</td>
<td>10</td>
</tr>
</tbody>
</table>

**Basic Concepts for Monterey Phoenix (MP) Behavioral Modeling**

- **Event** - any detectable action in system’s or environment’s behavior
- **Event trace** - set of events with two basic partial ordering relations, *precedence* (PRECEDES) and *inclusion* (IN)
- **Event grammar** - specifies the structure of possible event traces

---

**Innovations:**

- **Uniform Framework:** Describe behaviors and interactions of the system AND environment using the same framework
- **Leverage Small Scope Hypothesis:** Exhaustive search through all possible scenarios (up to the scope limit), expecting that most flaws in models can be demonstrated on small counterexamples
- **Separation of System Interaction from System Behavior:** Specify behavior of each system’s components separately from interactions between those components