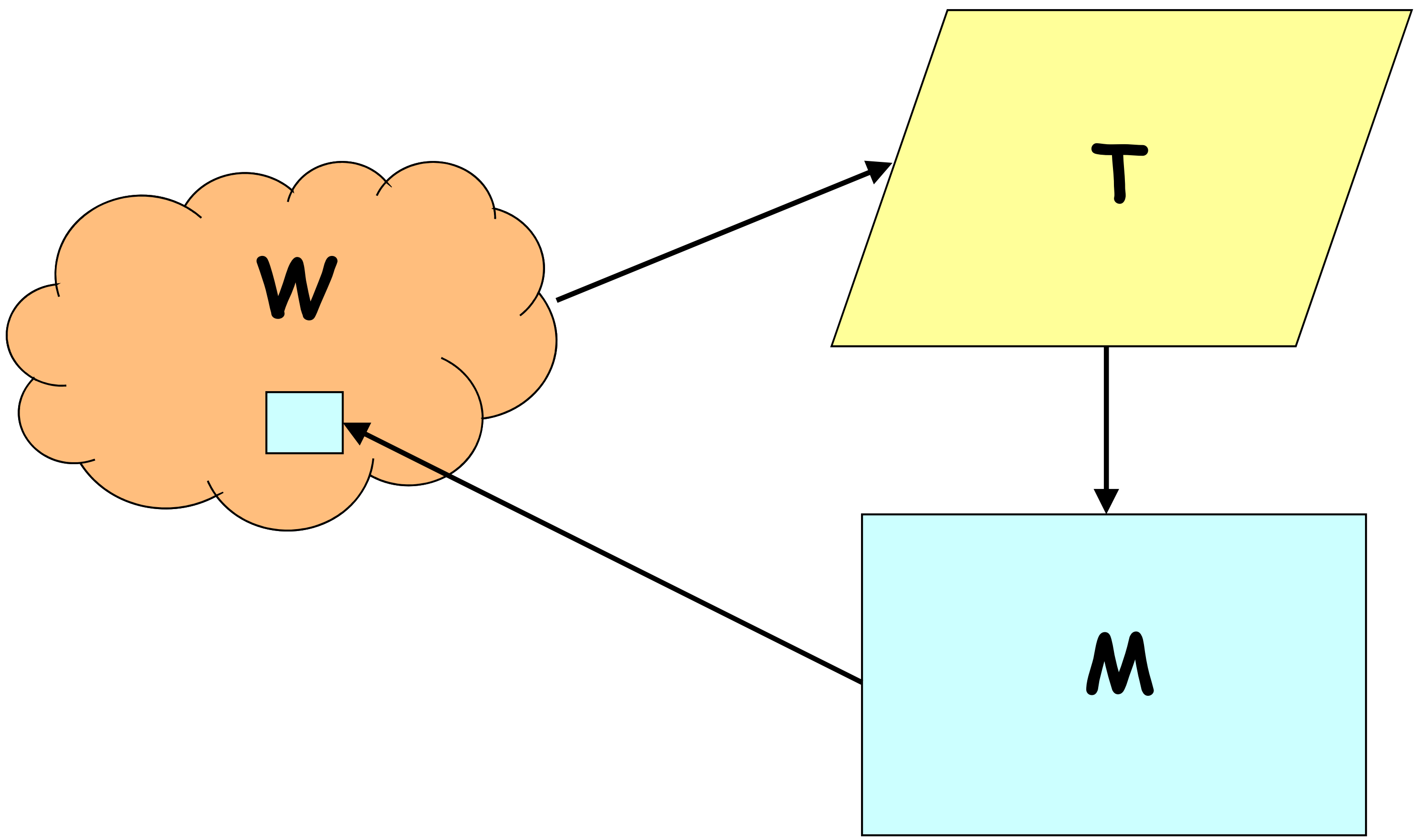
A Unifying Theoretical Foundation (or Platform) for Software Engineering

Dewayne E Perry
ACE 5.124
perry@mail.utexas.edu

Introduction

- I have a general theory about software engineering – it is made up of two basic endeavors:
  - Design
    - Of the problem
    - Of the solution
    - And includes a mundane manufacture
  - Evaluation
    - Of our problem, its solution, and the solution's utility
    - Of our evaluations themselves
- So lets consider two simple theories
  - Overly simple at this point, but for illustrative purposes
- See also:
  - "A Unifying Theoretical Foundation for Software Engineering" users.ece.utexas.edu/~perry/work/papers/DP-sede11.pdf
  - "A Theoretical Foundation for Software Engineering: A Model Calculus" – GTSE 2013 Proceedings

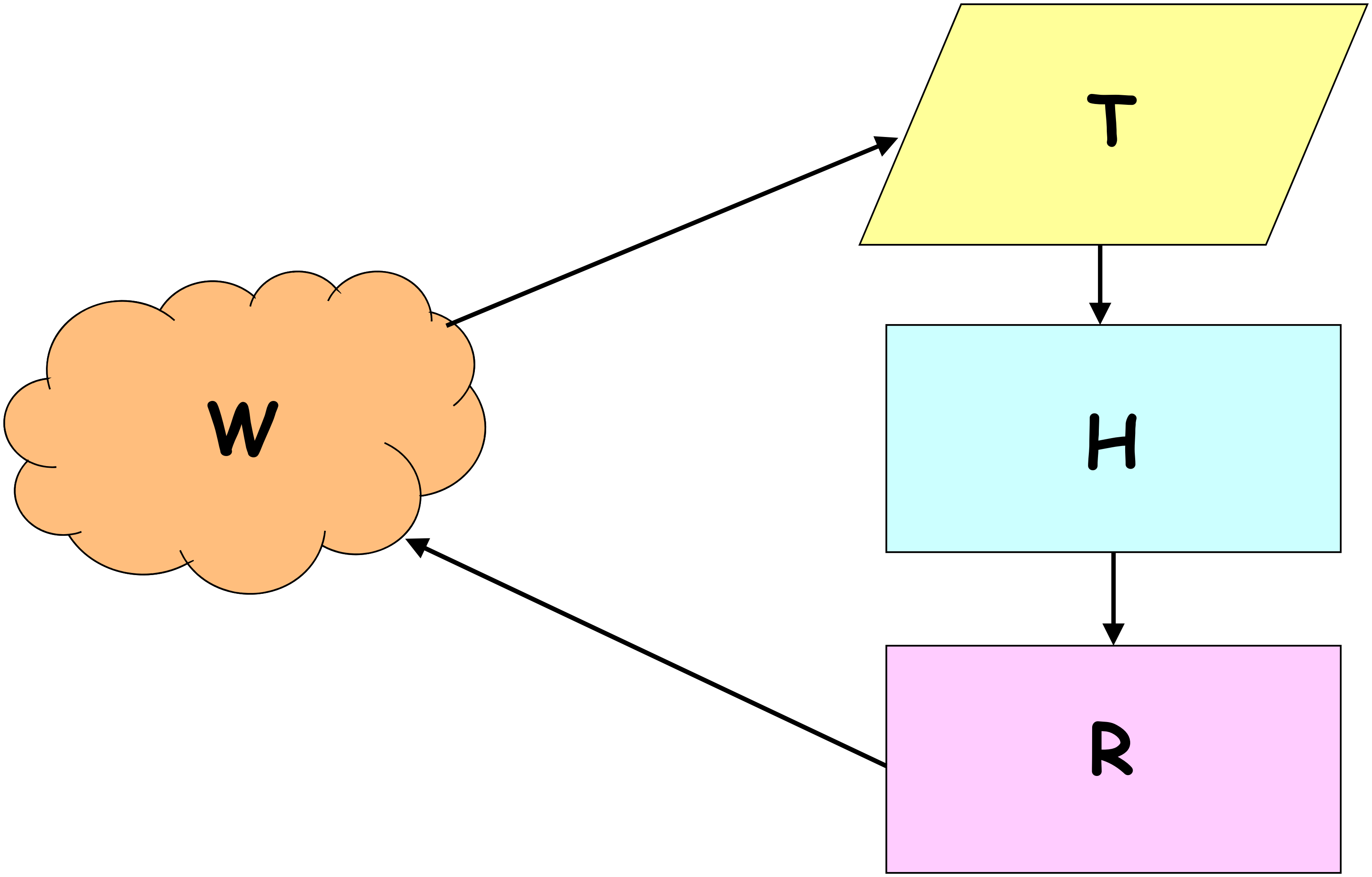
Theory D



Theory D

- Part of software engineering is an interpretation (or a model) of D (granted this is a very simple view of SE)
  - ★ The problem space that software engineering deals with is found in the world W
  - ★ The requirements we define to delineate that problem in the world is an interpretation of the theory T
  - ★ The system we build to satisfy the set of requirements (ie, the theory T) is an interpretation of the model M
  - ★ Eliciting the requirements from the world is our first process step in creating and building (ie, engineering) a software system
  - ★ Transforming those requirements into a system is the second process step in creating and building a software system
  - ★ Finally, the process of injecting the system into the world is the last step
  - ★ That injection into the world changes it, and we iterate.

Theory E



Theory E

- Part of software engineering is basically a continuing set of evaluations supporting and justifying design and process choices in D
  - The world W is the context for our evaluations – it is the source of our theories for various things related to our system
  - We build theories about these aspects of the world to be tested for confirmation or reformation
  - We do this by designing an hypothesis and a regimen (an experimental design – yes, back to D here ☺) to test some part of the theory
  - We follow processes for creating and adjusting theories, creating hypotheses (or evaluation questions), creating empirical designs/regimens, and executing and evaluating their results
- Evaluations can range from informal to formal

Theories D & E

- My theory of software engineering:
  - SE = D + E
  - But done in a specific way
  - E applied to D
  - Informally, we integrate E into D
- Lets consider simple models of D and E
  - First, define the models for D and E
  - Second, Apply E to D
  - Then, explore what that application means in terms of software engineering

Model of D

- **W** The world - more specifically, the relevant part of the world
- **T** The theory initiated by observation and abstraction
- **M** A model that satisfies the theory
- **W→T** Generate a theory: observe and abstract from the world (W) to create a theory (T)
- **T→M** From theory (T) create a model (M)
- **M*W→W** Inject the model (M) into the world (W) - Thereby changing the world

Model of E

- W — The world - more specifically, the relevant part of the world
- T — The theory initiated by observation and abstraction from the world
- H — An hypothesis to test the theory
- R — An regimen to test the hypothesis
- W → T — Generate a theory: observe and abstract from the world (W) to create a theory (T)
- T → H — From theory (T) generate an hypothesis (H)
- H → R — From hypothesis (H) generate an empirical evaluation to test it
- R * W → T — Apply R to W and reconcile T with reality

Theories D & E

- **What do we get from my approach:**
  - ★ Scientific elegance in creating larger more complex theories out of simpler theories
  - ★ Explain the complexity of software engineering and software engineering research in an elegant way.
  - ★ A theory modeling language and a calculus for composing models
- **Why my approach?**
  - ★ Description and understanding of what we do
  - ★ Provide a basis for exploring various approaches and what they entail
  - ★ To delineate the landscape of SE and RSE – lay out a taxonomic space
  - ★ To emphasize the importance of *explicating theories* in both software engineering design and empirical evaluation
  - ★ To emphasize the importance and extent of the *empirical part* of Software Engineering

Models and Applications

- What is a model – a tuple of a set of objects and a set of mappings
  - ★ < {objects}, {mappings} >
  - ★ Objects – eg, W, T, M, etc
    - Elements, components, entities, etc
  - ★ Mappings - W → T, R*W → T, etc
    - Transformations, generations, derivations, processes, etc
- Can treat models as
  - ★ Atomic (A) – ie, abstract away the internal structure
  - ★ Open structured (O) – ie, expose the internal structure
- Application: X:Y
  - ★ Basically restricts X to Y – ie, focuses X on Y
  - ★ A:O - yields n models where n is the number of objects + the number of mappings
  - ★ O:A – yields 1 model where the objects and mappings are restricted to A

Model Calculus - Operators

- Special symbols (in their order of precedence)
  - "+" a unary operator on objects that indicates 1 or more of the designated objects.
  - ":" a binary operator on models and model components that indicates a restriction of the left model or element to the right element or model
  - "*" a binary operator on objects that delineates an object in the Cartesian space of two objects. This can be thought of as functional application of the one object to the other yielding a specific object as its value.
  - "→" a binary operator that maps one object onto another.
  - Parentheses may be used to clarify the use of these operators.

Model Calculus - Mappings

- All possible mappings are possible
  - One to one mappings are indicated by A → B.
  - Many to one mappings are indicated in several different ways. For example, A * B → C, and A+ → B.
  - One to many mappings are indicated by A → B+ and A → B * C.
  - Many to many mappings are indicated by any combinations using "+" and "*" together with "→"

Model Calculus - Rules

- The following are the distribution rules among expressions about various operators.
  - ":" is both left and right distributive over models.
  - ":" is left distributive over "+", "*", and "→".
- Examples of the first distribution rule are above. Examples of the second are as follows (where EM denotes a model or a model element):
  - $(O^1 \rightarrow O^2):EM \quad = \quad O^1:EM \rightarrow O^2:EM$
  - $(O^1 * O^2 \rightarrow O^3):EM \quad = \quad O^1:EM * O^2:EM \rightarrow O^3:EM$
  - $(O^{1+} \rightarrow O^2):EM \quad = \quad (O^1:EM)+ \rightarrow O^2:EM$
- There is one rule about the operator "+" (which implies that "+" is left distributive) over "*" and "→". For example,
  - $(A \rightarrow B)+ \quad = A+ \rightarrow B+$
  - $(A * B)+ \quad = A+ * B+$

Applying E to D

- Consider evaluating D as atomic and E as open structured
  - E:D creates a new model of empirical evaluation focused specifically on D – ie, evaluating D as a whole

```
<
 {W:D, T:D, H:D, R:D},
 {
   (W → T):D          => W:D → T:D
   (T → H):D          => T:D → H:D
   (H → R):D          => H:D → R:D
   (R * W → T):D      => R:D * W:D → T:D
 }
>
```

  - That is, there are theories about D (derived from the world W:D) from which we create hypotheses and from those hypotheses we create regimens (ie, evaluations) – and we have the processes for doing just that with respect to D

Applying E to D

- Consider E to be abstract and D open structured
- Evaluating each object and mapping of D
  - E:W - evaluation of W (the world)
  - E:T - evaluation of T (the theory)
  - E:M - evaluation of M (the model)
  - E:(W → T) - evaluation of generating T from W
  - E:(T → M) - evaluation of transforming T into M
  - E:(M → W) - evaluation of injecting M into W
- We get 6 models of how to evaluate D, one for each object and one for each mapping
- To understand these models resulting from application
  - Consider E:m – since m is atomic, let E be open structured

Applying E to D

- **E:W - evaluation of W (the world)**
  - <{W:W,T:W, H:W, R:W},
    {W:W→T:W, T:W→H:W, H:W→R:W, R:W*W:W→T:W}>
  - Evaluation of (for example)
    - The users needs and desires
    - The world of "using"
    - The problem domain
- **E:T - evaluation of T (the theory)**
  - <{W:T,T:T, H:T, R:T},
    {W:T→T:T, T:T→H:T, H:T→R:T, R:T*W:T→T:T}>
  - Evaluation of (for example)
    - Theory completeness
    - Theory representativeness – of the user's needs/problems
    - Theory adequacy – how good is the theory for the model?

Applying E to D

- E:M - evaluation of M (the model)
  - <{W:M,T:M, H:M, R:M},
    {W:M→T:M, T:M→H:M, H:M→R:M, R:M*W:M→T:M}>
  - Evaluation of (for example)
    - Model adequacy
      - ✓ How well does the model represent the theory (white box)
      - ✓ How good is the model relative to the intent of the modeller (black box)
    - Model Utility – how useful is the model in the world
- E:(W→T) - evaluation of transforming W into T
  - <{W:(W→T), T:(W→T), H:(W→T), R:(W→T)},
    {W:(W→T)→T:(W→T), T:(W→T)→H:(W→T),
    H:(W→T)→R:(W→T),
    R:(W→T)*W:(W→T)→T: (W→T)}>
  - Evaluation of (for example)
    - The quality of the theory formation process

Applying E to D

E:(T→M) - evaluation of transforming T into M

- <{W:(T→M), T:(T→M), H:(T→M), R:(T→M)},
  {W:(T→M)→T:(T→M), T:(T→M)→H:(T→M),
  H:(T→M)→R:(T→M),
  R:(T→M)*W:(T→M)→T:(T→M)}>
- Evaluation of (for example)
  - The quality of the model development process

❖ E: (M→W) - evaluation of injecting M into W

- <{W:(M→W), T:(M→W), H:(M→W), R:(M→W)},
  {W:(M→W)→T: (M→W), T:(M→W)→H:(M→W),
  H:(M→W)→R:(M→W),
  R:(M→W)*W:(M→W)→T:(M→W)}>
- Evaluation of (for example)
  - The quality of the model deployment process

E:D - The Full View

E:D	E:W	E:T	E:M
W:D	W:W	W:T	W:M
T:D	T:W	T:T	T:M
H:D	H:W	H:T	H:M
R:D	R:W	R:T	R:M
(W → T):D	W:W→T:W	W:T→T:T	W:M→T:M
(T → H):D	T:W→H:W	T:T→H:T	T:M→H:M
(H → R):D	H:W→R:W	H:T→R:T	H:M→R:M
(R * W → T):D	R:W*W:W→T:W	R:T*W:T→T:T	R:M*W:M→T:M

E:(W → T)	E:(T → M)	E:(M → W)
W:(W→T)	W:(T→M)	W:(M→W)
T:(W→T)	T:(T→M)	T:(M→W)
H:(W→T)	H:(T→M)	H:(M→W)
R:(W→T)	R:(T→M)}	R:(M→W)
W:(W→T)→T:(W→T)	W:(T→M)→T:(T→M)	W:(M→W)→T: (M→W)
T:(W→T)→H:(W→T)	T:(T→M)→H:(T→M)	T:(M→W)→H:(M→W)
H:(W→T)→R:(W→T)	H:(T→M)→R:(T→M)	H:(M→W)→R:(M→W)
R:(W→T)*W:(W→T)→T: (W→T)	R:(T→M)*W:(T→M)→T:(T→M)	R:(M→W)*W:(M→W)→T:(M→W)

So Far

- 2 simple theories and their models which together comprise SE
- Applied E to D to *get* E:D – a more complex theory and model that delineates the empirical side of SE
- Claim: SE is D & E:D & E:(E:D) [see below]
- E:D lays out a taxonomic space
  - For any application, the space is a matrix m x n
  - In the case of E:D – 8 x 6, of E:(E:D) – 8 x 8 x 6
  - So the more complex the theory and model (which we will propose below), the larger the taxonomic space

- Lets backtrack a bit and look deeper at the issues about theories and models

Theories and Models

- Terms often used in a variety of ways
  - Informally, interchangeably
- Want to use in a very specific way
  - A theory is a description (an abstract entity)
  - Reified, represented, satisfied, etc by a model (a concrete entity)
- Derived from Turski and Maibaum [TM87]
  - "A specification is rather like a natural science theory of the application domain, but seen as a theory of the corresponding program it enjoys an unmatched status: it is truly a postulative theory, the program is nothing more than an exact embodiment of the specification"
- I want a theory to be broader than a specification and less formal
  - A specification is in fact part of the model, not the theory

Interesting Differences

- Between natural, behavioral & design sciences
- Natural sciences
  - Understand natural phenomena
  - Theoretical basis for prediction
  - Theories have to be testable
  - Testing is done in the physical world
  - Physical world provides *hard constraints* on theories
  - Basis for invention/engineering
  - Educational basis
    - Stream of experimental work in laboratory components of basic science courses
    - Explicitly recognize the need for both theoretical and experimental enterprises

Interesting Differences

- Behavioral sciences
  - Understand human and societal phenomena
  - Theoretical basis for predictions and interventions
    - Basis for prediction
    - Basis for intervening and changing the world
      - SE also has this property
  - Theories have to be testable
  - Testing is done in the behavioral world
  - Behavioral world provides *probabilistic constraints*
  - Basis for therapy, education, policy, motivation, etc
  - Educational basis:
    - Experimental design and statistics courses as both undergraduate and graduate
    - Explicitly recognize the need for both theoretical and experimental enterprises

Interesting Differences

- Design Sciences (sciences of the artificial)
  - Understand artificial phenomena
  - Theoretical basis for artificial interventions/improvements
  - Theories have to be testable
  - Testing is done in at least part in an artificial world
  - Artificial world provides *selectable and malleable constraints*
  - Basis for design invention and purpose-based transformations
    - Artifacts with desired properties
    - Transformations to achieve desired goals
  - Educational basis
    - Unfortunately, haphazard and ad hoc
    - Lack of appreciation for experimental aspects

Source of Theories

- Scientific Theory
  - Based on observations about the world
    - Some observations sets very old but still used/useful
  - Changed on the basis of
    - New interpretations of observations
    - New observations
- Legal Theory
  - Based on decisions about the world
  - Changed on the basis of
    - New interpretations of decisions
    - New decisions
- Normative Theory
  - Based on a system of philosophical tenets about what is good and bad
  - Changed on the basis of
    - New inferences from those tenets
    - New interpretations of them
    - New tenets

Source of Theories

- Theories in design disciplines – a combination of the three
  - Based on
    - Observations about the world
    - Decisions about the world
    - Judgments about the world
  - Changed on the basis of
    - New interpretations of the observations or decision
    - New observations or decisions
    - New inferences, interpretations, or tenets

Structure of Theories

- Two different theory structures [Markus & Robey]
  - Variance
    - Theoretical structure is a set of laws about
      - Interactions
      - Relationships
    - Given a variation in A, what other things can account for that variation
  - Process
    - Theoretical structure is a temporal ordering of activities, steps or events
- We have both theoretical structures in the sciences of the artificial

Uses of Theories

- 5 distinct uses [derived from Gregor] – all occur in SE
  - Description
    - In terms of constructs, properties, relationships and boundaries
    - Intended to be complete
  - Prescription
    - Constraints on it constructs, properties, relationships, boundaries
    - Intent is to emphasize critical/crucial aspects of the theory
  - Explanation
    - How why when things occur based on causality and demonstration
    - Intent: deeper understanding and insight
  - Prediction
    - Based on necessary and sufficient conditions
    - Determines when it will or will not happen
  - Action
    - Principles, techniques and methods for enabling phenomena for achieving a desired goal, or designing/constructing an artifact
- We have all these uses in SE practice and SE research

Context for Design Sciences

- Physical world provides hard constraints on theories
- Behavioral world provides probabilistic constraints
- Technological world provides selectable constraints
- Intellectual context provides malleable constraints

D:D Theory

- Composing/Applying D with/to itself gives us D:D- a theory and model for creating and evolving a D (a theory and model)
- We observe and abstract some specific part (that of D) of the world and create a theory of
  - what the world is like and what is important
  - what form theory should take
  - what form the model should take
  - the processes of creating the theory and a model and injecting it into the world
- From that theory we create a usable model of
  - what the world is like and what is important
  - what form theory should take
  - what form the model should take
  - the processes of creating the theory and a model and injecting it into the world

D:D Theory

- From the world of D we derive a theory about D
- From this theory of D we then derive a model of D
- We then inject this model of D into the world of D, thereby changing that world W:D
- These changes in the world may then lead to adjustments and extensions to the original theory about D
- This then leads to further changes in the model of D and subsequently, again, the world of D

D:D Model

- Considering D as an open structured model applied to an abstract model we get the following model (in which the elements of D are restricted to the abstract model D)
  - W:D
  - T:D
  - M:D
  - (W → T):D => W:D → T:D
  - (T → M):D => T:D → M:D
  - (M → W):D => M:D → W:D
- There are theories about D (T:D) derived from the world (W:D) from which we create models of D (M:D) – and we have the processes for doing just that with respect to D

Digging Deeper into D:D

- Viewing D now as an open structure model we can apply each object and mapping of D to the elements and mappings of D and gain a deeper understanding of what the restricted focus actually means

- W:D – the world related to D
  - W:W, W:T, W:M,
  - W:(W→T), W:(T→M), W:(M→T)
- T:D – the theory of D
  - T:W, T:T, T:M,
  - T:(W→T), T:(T→M), T:(M→W)
- M:D – the model of D
  - M:W, M:T, M:M,
  - M:(W→T), M:(T→M), M:(M→W)

Digging Deeper into D:D

- (W→T):D => W:D → T:D
  - W:W → T:W
  - W:T → T:T
  - W:M → T:M
  - W:(W→T) → T:(W→T)
  - W:(T→M) → T:(T→M)
  - W:(M→W) → T:(M→W)
- (T→M):D => T:D → M:D
  - T:W → M:W
  - T:T → M:T
  - T:M → T:M
  - T:(W→T) → M:(W→T)
  - T:(T→M) → M:(T→M)
  - T:(M→W) → M:(M→W)
- (M→W):D = M:D → W:D
  - M:W → W:W
  - M:T → W:T
  - M:M → W:M
  - M:(W→T) → W:(W→T)
  - M:(T→M) → W:(T→M)
  - M:(M→W) → W:(M→M)

Applying D to D

- D:D yields 6 models (one for each object and mapping)
  - D:W - designing the problem space
  - D:T - designing what theories look like
  - D:M - designing what models look like
  - D:(W → T) - designing the process of deriving theories
  - D:(T → M) - designing the process of deriving models
  - D:(M → W) - designing the process of deployment
- We get 6 models of how to design and derive D, one for each object and one for each mapping
- To understand these models resulting from application
  - Let D be open structured

Applying D to D

- D:W - design of W (the world)
  - <{W:W,T:W, M:W},
    {W:W→T:W, T:W→M:W, M:W→W:W}>
  - Design of (for example)
    - The users needs and desires
    - The world of “using”
    - The problem domain
- D:T - design of T (the theory)
  - <{W:T,T:T, M:T},
    {W:T→T:T, T:T→M:T, M:T→W:T}>
  - Design of (for example)
    - What a theory should be
    - What constitutes theory completeness, representativeness, and adequacy

Applying D to D

- D:M - design of M (the model)
  - <{W:M,T:M, M:M},
    {W:M→T:M, T:M→M:M, M:M→W:M}>
  - Design of (for example)
    - What a model should look like
    - What constitutes model completeness, adequacy and utility
- D:(W→T) – design of the process of deriving T from W
  - <{W:(W→T), T:(W→T), M:(W→T)},
    {W:(W→T)→T:(W→T), T:(W→T)→M:(W→T),
    M:(W→T)→W:(W→T)}>
  - Design of (for example)
    - What the theory formation process should be

Applying D to D

- D:(T→M) – design of the process of deriving M from T
  - ★ <{W:(T→M), T:(T→M), M:(T→M)},
    {W:(T→M)→T:(T→M), T:(T→M)→M:(T→M),
    M:(T→M)→W:(T→M)}>
  - ★ Design of (for example)
    - ➢ What the model development process should be
- ❖ D: (M→W) – design of the process of injecting M into W
  - ★ <{W:(M→W), T:(M→W), M:(M→W)},
    {W:(M→W)→T: (M→W), T:(M→W)→M:(M→W),
    M:(M→W)→W:(M→W)}>
  - ★ Evaluation of (for example)
    - ➢ What the model deployment process should be

The D:D Matrix

D:D	D:W	D:T	D:M
W:D	W:W	W:T	W:M
T:D	T:W	T:T	T:M
M:D	M:W	M:T	M:M
(W → T):D	W:W→T:W	W:T→T:T	W:M→T:M
(T → M:D	T:W→M:W	T:T→M:T	T:M→M:M
(M → W):D	M:W→W:W	M:T→W:T	M:M→W:M

D:(W → T)	D:(T → M)	D:(M → W)
W:(W→T)	W:(T→M)	W:(M→W)
T:(W→T)	T:(T→M)	T:(M→W)
M:(W→T)	M:(T→M)	M:(M→W)
W:(W→T)→T:(W→T)	W:(T→M)→T:(T→M)	W:(M→W)→T: (M→W)
T:(W→T)→M:(W→T)	T:(T→M)→M:(T→M)	T:(M→W)→M:(M→W)
M:(W→T)→W:(W→T)	M:(T→M)→W:(T→M)	M:(M→W)→W:(M→W)

SE Research – A Model for D:D

- Research in software development (software engineering – SE) is a model for D:D
  - The problem spaces concerning SE research, SE theories, SE models, and SE processes constitute a model for W
  - SE research theories about what form W, T, M, and the various D transformations should take are models of T
  - SE research models about what form W, T, M, and the various D transformations should take are models of M
  - The processes we as SE researchers use are models of D:D transformations
- Similarly, research in creating project management plans, empirical instruments, and empirical evaluations are models of D:D

Examples for D:D

- The world of software systems development
  - Some problems are just too hard – unsolvable
  - Vincenti – some problems require only normal design, others require radical design (ie, we don't know whether it will work or not)
  - Brooks: Plan to throw one away – we don't know how to build a system until we have actually built it at least once
- Theories of software systems development
  - Multiple viewpoints (a theory about theory – ie, a T:T):
    - Different stakeholders have different views of what needs to be done and needs to be part of the requirements (T:T, T:(W→T))
    - Conflict resolution and how it is accomplished is a critical part of the requirements (T:T, T:(T→T))
  - Perry/Wolf model of software architecture
    - A theory about what part of a model of software systems should be like

Examples for D:D

- Other theories of what models should look like
  - Structured programming
  - Object oriented programming
  - Aspect-oriented programming
  - Top-down, bottom-up design
  - System should reflect the shape of the problem
  - Etc
- Theories of how to develop software systems
  - Royce's waterfall model
  - Boehm's spiral model
  - Extreme/agile development
  - Wirth and others development by refinement
  - Batory's feature-oriented development

Examples for D:D

- Models of software development
  - Design methods
    - Functional decomposition
    - Dataflow decomposition
    - Object oriented design
    - Etc
  - Architecture derivation
    - van Lamsweerde – KAOS via agents to architecture
    - Brandozzi/Perry – KAOS via goals to architecture
  - Technology transfer (M→W) [both examples from hyperCode]
    - Seamless integration into existing processes
    - Integrate development with research

So Far

- D:D is the design part of software engineering research
  - Have laid out the taxonomic space for D:D
  - Have claimed that SE Research, at least the design part of it, is a model of D:D

- That leaves us then to address the empirical evaluation side of software engineering research
  - E:(D:D) is E applied to D:D analogous to what we did for SE itself, E:D
  - Note that the taxonomic space for D:D is 6 x 6 = 36
  - The taxonomic space for our currently simple E:(D:D) is
    - 8 x 6 x 6 = 288
    - A very large space, even for our overly simple models

Applying E to D:D

- **Consider evaluating D:D as atomic and E as open structured**
  - ★ E:(D:D) creates a new model of empirical evaluation focused specifically on D:D – ie, evaluating D:D as a whole

```
<
  {W:(D:D), T:(D:D), H:(D:D), R:(D:D)},
  {
    (W → T):(D:D)          => W:(D:D) → T:(D:D)
    (T → H):(D:D)          => T:(D:D) → H:(D:D)
    (H → R):(D:D)          => H:(D:D) → R:(D:D)
    (R * W → T):(D:D)      => R:(D:D) * W:(D:D) → T :(D:D)
  }
>
```

  - ★ That is, there are theories about D:D (derived from the world W:(D:D)) from which we create hypotheses and from those hypotheses we create regimens (ie, evaluations) – and we have the processes for doing just that with respect to D:D

Applying E to D:D

- Consider E to be abstract, D^1 open structured, and D^2 abstract
- Evaluating each object and mapping of D^1:D^2
  - E:(W:D) - evaluation of W:D (the world of D)
  - E:(T:D) - evaluation of T:D (the theory of D)
  - E:(M:D) - evaluation of M:D (the model of D)
  - E:(W:D → T:D) - evaluation of transforming W:D into T:D
  - E:(T:D → M:D) - evaluation of transforming T:D into M:D
  - E:(M:D → W:D) - evaluation of injecting M:D into W:D
- We get 6 models of how to evaluate D^1:D^2, one for each object and one for each mapping of D^1:D^2
- To understand these models resulting from application
  - Let E be open structured

Applying E to D:D

- E:(W:D) - evaluation of W:D (the world of D)
  - < {W:(W:D), T:(W:D), H:(W:D), R:(W:D)}, {W:(W:D)→T:(W:D), T:(W:D)→H:(W:D), H:(W:D)→R:(W:D), R:(W:D)*W:(W:D)→T:(W:D)} >
  - ★ Evaluation of (for example)
    - The users needs and desires about D
    - The world of "using" D
    - The problem domain of D
- E:(T:D) - evaluation of T:D (the theory of D)
  - < {W:(T:D) ,T:(T:D), H:(T:D), R:(T:D)}, {W:(T:D)→T:(T:D), T:(T:D)→H:(T:D), H:(T:D)→R:(T:D), R:(T:D)*W:(T:D)→T:(T:D)} >
  - ★ Evaluation of (for example)
    - Theory completeness
    - Theory representativeness – of the user's needs/problems
    - Theory adequacy – how good is the theory for the model?

Applying E to D:D

- **E:(M:D) - evaluation of M (the model of D)**
  - < {W:(M:D),T:(M:D), H:(M:D), R:(M:D)},
    {W:(M:D)→T:(M:D), T:(M:D)→H:(M:D), H:(M:D)→R:(M:D),
    R:(M:D) *W:(M:D) →T:(M:D)} >
  - ★ Evaluation of (for example)
    - ➢ Model of D adequacy
      - ✓ How well does the M:D represent T:D (white box)
      - ✓ How good is the M:D relative to the intent of the modeller (black box)
    - ➢ Model of D utility – how useful is the M:D in W:D
- **E:(W:D→T:D) - evaluation of transforming W:D into T:D**
  - ➢ <{W:(W:D→T:D), T:(W:D→T:D), H:(W:D→T:D), R:(W:D→T:D)},
    {W:(W:D→T:D)→T:(W:D→T:D), T:(W:D→T:D)→H:(W:D→T:D),
    H:(W:D →T:D)→R:(W:D →T:D),
    R:(W:D→T:D)*W:(W:D→T:D)→T:(W:D→T):D}>
  - ★ Evaluation of (for example)
    - ➢ The quality of the theory of D formation process

Applying E to D:D

E:(T:D→M:D) - evaluation of transforming T:D into M:D

<{W:(T:D→M:D), T:(T:D→M:D), H:(T:D→M:D),
R:(T:D→M:D)},
{W:(T:D→M):D→T:(T:D→M:D), W:(T:D→M:D)→H:(T:D→M:D),
H:(T:D→M:D)→R:(T:D→M:D),
R:(T:D→M:D)*W:(T:D→M:D)→T:(T:D→M:D)}>

- ★ Evaluation of (for example)
  - The quality of the model formation process

- E: (M→W) - evaluation of injecting M:D into W:D

<{W:(M:D→W:D), T:(M:D→W:D), H:(M:D→W:D),
R:(M:D→W:D)},
{W:(M:D→W:D)→T: (M:D→W:D), T:(M:D→W:D)→H:(M:D→W),
H:(M:D→W:D)→R:(M:D→W:D),
R:(M:D→W:D)*W:(M:D→W:D)→T:(M:D→W:D)}>

- ★ Evaluation of (for example)
  - The quality of the model deployment process

The E:(D:D)Matrix - Example T:(D:D)

T:(D:D)	D:W	D:T	D:M
W:D	T:(W:W)	T:(W:T)	T:(W:M)
T:D	T:(T:W)	T:(T:T)	T:(T:M)
M:D	T:(M:W)	T:(M:T)	T:(M:M)
(W → T):D	T:(W:W→T:W)	T:(W:T→T:T)	T:(W:M→T:M)
(T → M:D	T:(T:W→M:W)	T:(T:T→M:T)	T:(T:M→M:M)
(M → W):D	T:(M:W→W:W)	T:(M:T→W:T)	T:(M:M→W:M)

D:(W → T)	D:(T → M)	D:(M → W)
T:(W:(W→T))	T:(W:(T→M))	T:(W:(M→W))
T:(T:(W→T))	T:(T:(T→M))	T:(T:(M→W))
T:(M:(W→T))	T:(M:(T→M))	T:(M:(M→W))
T:(W:(W→T)→T:(W→T))	T:(W:(T→M)→T:(T→M))	T:(W:(M→W)→T: (M→W))
T:(T:(W→T)→M:(W→T))	T:(T:(T→M)→M:(T→M))	T:(T:(M→W)→M:(M→W))
T:(M:(W→T)→W:(W→T))	T:(M:(T→M)→W:(T→M))	T:(M:(M→W)→W:(M→W))

Models of E:(D:D)

- Comparison of the van Lamsweerde's and the Brandozzi/Perry methods of transforming KAOS requirements into architecture descriptions or prescriptions
  - Vanderveken, Jani, and Perry – ECSA 2005
    - van Lamsweerde generated a detailed architecture description
    - Brandozzi/Perry generated a high-level architecture prescription
    - van Lamsweerde easier to get started, harder to finish
    - Brandozzi/Perry harder to get started, easier to finish
- Evaluation of semantic conflict analysis as a mechanism for predicting faults
  - Shao, Khurshid, & Perry – ICSM 2007
    - better at predicting semantic conflicts in highly parallel changes in adaptive changes than less parallel forms of changes in fixing faults or improvements

Evaluating Evaluations – E:E

- Consider evaluating E^1 as atomic and E^2 as open structured
  - E^2 : E^1 creates a new model of empirical evaluation focused specifically on E^1 – ie, evaluating E^1 as a whole

```
<
 {W:E, T:E, H:E, R:E},
 {
   (W → T):E          => W:E → T:E
   (T → H):E          => T:E → H:E
   (H → R):E          => H:E → R:E
   (R * W → T):E      => R:E * W:E → T:E
 }
>
```

  - That is, there are theories about E (derived from the world W:E) from which we create hypotheses and from those hypotheses we create regimens (ie, evaluations) – and we have the processes for doing just that with respect to E

Applying E to E

- Consider E^2 to be abstract and E^1 open structured
- Evaluating each object and mapping of E^1
  - ★ E:W - evaluation of W (the world)
  - ★ E:T - evaluation of T (the theory)
  - ★ E:H - evaluation of H (the hypothesis)
  - ★ E:R - evaluation of R (the regimen/evaluation)
  - ★ E:(W → T) - evaluation of deriving T from W
  - ★ E:(T → H) - evaluation of deriving H from T
  - ★ E:(H → R) - evaluation of deriving R from H
  - ★ E:(R * W → T) - evaluation of applying R to W and reconciling T with reality
- We get 8 models of how to evaluate E, one for each object and one for each mapping
- To understand these models resulting from application
  - ★ Let E be open structured in the following

Applying E to E

- E:W - evaluation of W (the world of E)
  - <{W:W,T:W, H:W, R:W},
    {W:W→T:W, T:W→H:W, H:W→R:W, R:W*W:W→T:W}>
  - Evaluation of (for example)
    - Appropriateness of E relative to the world W of possible Es
- E:T - evaluation of T (the theory of E)
  - <{W:T,T:T, H:T, R:T},
    {W:T→T:T, T:T→H:T, H:T→R:T, R:T*W:T→T:T}>
  - Evaluation of (for example)
    - Theory completeness and consistency
    - Theory appropriateness and adequacy

Applying E to E

- E:H - evaluation of M (the hypothesis)
  - <{W:H,T:H, H:H, R:H},
    {W:H→T:H, T:H→H:H, H:H→R:H, R:H*W:H→T:H}>
  - Evaluation of (for example)
    - The appropriateness of the hypothesis H relative to theory T
    - Construct validity of the hypothesis H
- E:R - evaluation of R (the regimen)
  - <{W:R,T:R, H:R, R:R},
    {W:R→T:R, T:R→H:R, H:R→R:R, R:R*W:R→T:R}>
  - Evaluation of (for example)
    - The appropriateness of regimen R relative to hypthesis H and theory T
    - Construct validity of regimen R
    - Internal validity of regimen R

Applying E to E

- E:(W→T) - evaluation of transforming W into T
  - <{W:(W→T), T:(W→T), H:(W→T), R:(W→T)},
    {W:(W→T)→T:(W→T), T:(W→T)→H:(W→T),
    H:(W→T)→R:(W→T),
    R:(W→T)*W:(W→T)→T: (W→T)}>
  - Evaluation of (for example)
    - The quality of the theory derivation process for E

E:(T→H) - evaluation of deriving H from T
  - <{W:(T→H), T:(T→H), H:(T→H), R:(T→H)},
    {W:(T→H)→T:(T→H), T:(T→H)→H:(T→H),
    H:(T→H)→R:(T→H),
    R:(T→H)*W:(T→H)→T:(T→H)}>
  - Evaluation of (for example)
    - The quality of the hypothesis derivation process for E

Applying E to E

- E: (H→R) - evaluation of deriving R from H
  - <{W:(H→R), T:(H→R), H:(H→R), R:(H→R)},
    {W:(H→R)→T:(H→R), T:(H→R)→H:(H→R),
    H:(H→R)→R:(H→R),
    R:(H→R)*W:(H→R)→T:(H→R)}>
  - Evaluation of (for example)
    - The quality of the regimen derivation process for E
    - Construct and internal validity
- E: (R*W→T) - evaluation of applying R and reconciling T with reality
  - <{W:(R*W→T), T:(R*W→T), H:(R*W→T), R:(R*W→T)},
    {W:(R*W→T)→T:(R*W→T), T:(R*W→T)→H:(R*W→T),
    H:(R*W→T)→R:(R*W→T),
    R:(R*W→T)*W:(R*W→T)→T:(R*W→T)}>
  - Evaluation of (for example)
    - The quality of the process for applying R and reconciling T with reality
    - External validity

E:E - The Full View

E:E	W	T	H	R	W → T
W	W:W	W:T	W:H	W:R	W:(W→T)
T	T:W	T:T	T:H	T:R	T:(W→T)
H	H:W	H:T	H:H	H:R	H:(W→T)
R	R:W	R:T	R:H	R:R	R:(W→T)
W → T	W:W→T:W	W:T→T:T	W:H→T:H	W:R→T:R	W:(W→T)→T:(W→T)
T → H	T:W→H:W	T:T→H:T	T:H→H:H	T:R→H:R	T:(W→T)→H:(W→T)
H → R	H:W→R:W	H:T→R:T	H:H→R:H	H:R→R:R	H:(W→T)→R:(W→T)
R * W → T	R:W*W:W→T:W	R:T*W:T→T:T	R:H*W:H→T:H	R:R*R:M→T:R	R:(W→T)*W:(W→T)→T:(W→T)

T → H	H → R	R * W → T
W:(T→H)	W:(H→R)	W:(R*W→T)
T:(T→H)	T:(H→R)	T:(R*W→T)
H:(T→H)	H:(H→R)	H:(R*W→T)
R:(T→H)}	R:(H→R)	R:(R*W→T)
W:(T→H)→T:(T→H)	W:(H→R)→T:(H→R)	W:(R*W→T)→T:(R*W→T)
T:(T→H)→H:(T→H)	T:(H→R)→H:(H→R)	T:(R*W→T)→H:(R*W→T
H:(T→H)→R:(T→H)	H:(H→R)→R:(H→R)	H:(R*W→T)→R:(R*W→T)
R:(T→H)*W:(T→H)→T:(T→H)	R:(H→R)*W:(H→R)→T:(H→R)	R:(R*W→T)*W:(R*W→T)→T:(R*W→T)

Where are We?

- Two simple theories (D and E) and their models have been sufficient to lay out the space of SE and SE Research
  - SE -- D, E:D and E:(E:D)
  - SE Research – D:D, E:(D:D), E:(E:(D:D))
    - I leave E:(E:(D:D)) as an exercise for the reader
  - Where do E and E:E fit in?
- Even with the simplest of theories and models the compositions of these theories and models becomes very complex
  - E:(E:D) has 384 objects and mappings
  - E:(D:D) has 288 objects and mappings
  - E:(E:(D:D) has 2304 objects and mappings
- Not Surprising – Brooks' primary essential characteristic of software systems is *complexity*

More Complex Models

- Iterative D (ID) – extend D by adding 3 new mappings
  - T → T -- refine T
  - M → M -- refine M
  - M → T -- adjust T to better reflect M
- Refine ID's M = A (architecture), D (design), and C (code)
  - M is replaced by objects A, D, and C
  - Add a new object S (system) and a mapping C → S, and replace M → W with S → W
  - T → M is replaced with T → A, T → D, T → C
  - M → T is replaced with A → T, D → T, C → T
  - M → M is replaced with
    - A → D, D → C
    - A → A, D → D, C → C
    - C → A, D → A, C → D
  - RID now has 6 objects and 21 mappings –
    - much more complex
    - Compositions even more so

More Complex Models

- Empirical evaluations can be much more complex in practice involving beyond our simple theory and model
  - ★ Elaborate instruments
  - ★ Independent and dependent variables
  - ★ Analyses of the application of the regimens, and the instrumentation used to manipulate independent variables and capture the values of the dependent variables
  - ★ Iteration refining hypotheses, regimens, variables, instruments
- May need theories and models of various complexities, depending on where and how we want to use them
  - ★ For example, the theory of E^1 in $E^1{:}E^2$ might be a different theory of evaluation (perhaps even simpler) than our initial E.

Motivations

- Create a unifying theoretical basis for SE and SE research
  - ★ Goes beyond this – a unifying theoretical basis for design sciences
- Emphasize the centrality and criticality of empirical evaluations in SE and SE research
  - ★ Understanding of empirical considerations is particularly weak in SE research – needs serious improvement
  - ★ A more explicit, systematic and deeper approach to empirical evaluations in SE would improve systems significantly
    - Interestingly, some of the latest *salvations du jour* implicitly emphasize some empirical evaluations – should be explicit
- Emphasize the centrality and criticality of theory in both SE and SE research
  - ★ If there at all, implicit – needs to be explicit and central

Summary

- Small, simple theories D and E form the basis for laying out a very rich space and an underlying theoretical foundation for SE, SE research, and other design disciplines
  - Compose D and E into more complex theories to extend and illuminate the space for design disciplines
- Useful properties
  - Regularity among the various theories
  - Levels of abstraction (stratification) within the composed theories providing
    - Intuitive high level abstractions
    - Explicit low level detailed abstractions

Future Work

- Explore D:E and D:(E:E) and hence D:(E:D), D:(E:(D:D))
  - Designs for evaluating technology and design decisions
    - Especially, constructs and construct validity issues for technology and design decisions
    - We intuitively know X is better than Y, but are seriously deficient in appropriate observable and measureable constructs
  - Adapting physical and behavioral designs for use in evaluating SE issues.
  - Explore the problems and issues in using computers in empirical evaluations as both subjects and instruments
- Explore the utility of various levels of theory complexity
- Do an extensive literature categorization
  - To validate my theory and model composition approach
  - Illuminate areas that need to be addressed in SE and SE research

Ultimate Goals

- For this course:
  - Relate papers evaluated to my taxonomic space
  - Place project within that taxonomic space as well
- For my continued research in a unifying foundation for SE
  - Further explore and explicate the spaces delineated
  - Populate these spaces with examples for SE and empirical literature
    - To delineate examples of these spaces
    - To provide a validation of the utility of this approach
  - Monographs and papers to illustrate the results of this research
  - Perhaps a grad course to go deeper into this research
- Hope: the field recognizes the critical importance of the need for explicit delineation of the theory underlying our work and the need for systematic and rigorous empirical evaluation