Characterizing Process Variation

Borislava I. Simidchieva and Leon J. Osterweil

Laboratory for Advanced Software Engineering Research
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
Introduction

- A process model defines the coordination of agents performing activities using resources and artifacts
- Careful study of the model can help to identify improvements to the real-world process it reflects
- Complex real-world processes exhibit variation
- Such processes may be accommodated better by a family of process models
Approach

- Formally characterize different variation relations
- Reason at two different levels of abstraction
- Strive for improved *generation, analysis, and navigation*

**Problem Space**
- Driven by the variation needs indicated in the requirements specification
- Representation-independent

**Solution Space**
- Informed by the strengths and weaknesses of different implementation approaches
- Representation-specific
Approach

- Formally characterize different variation relations
- Reason at two different levels of abstraction
- Strive for improved *generation, analysis, and navigation*

**Problem Space**

- Functional variation
- Functional invariance
- Goal invariance
- Robustness variation
- Performance variation
- Interaction-based variation
- Agent variation

**Solution Space**

- Induction
  - Well-formedness constraints
- Structural transformation
- Agent behavior change
- Nominal flow change
- Exceptional flow change
- Agent behavior to process structure
- Process structure to agent behavior

*Characterizing Process Variation*, Simidchieva and Osterweil

ICSE NIER Track, May 25 2011
Advantages

Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications

2. **Analysis** of an entire process family at once
   - Reasoning about all variants collectively to determine if they meet certain properties in dimensions such as security, privacy, safety and correctness

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families
Advantages

Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications

2. **Analysis** of an entire process family at once
   - Reasoning about all variants collectively to determine if they meet certain properties in dimensions such as security, privacy, safety and correctness

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families
Advantages

Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications

---

**Legend:**
- Dataflow
- Artifacts
- Generation activities
- Libraries

---

**Process Common Core**

**Process Goal Specification**

**Task elaborations library**

**Coordination Model Generation**

**Functional Variation Process Variant**

---

*Characterizing Process Variation*, Simidchieva and Osterweil
Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications
Advantages

Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications

2. **Analysis** of an entire process family at once
   - Reasoning about all variants collectively to determine if they meet certain properties in dimensions such as security, privacy, safety and correctness

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families
Explicit modeling of process variation may help with:

2. **Analysis** of an entire process family at once
   - Reasoning about all variants collectively to prove safety and correctness properties

![Diagram with symbols and text describing analysis engine and variants.](image-url)
Explicit modeling of process variation may help with:

1. **Generation** of new variants
   - Creating a new variant based on pre-specified variation relations and known requirements and architecture specifications

2. **Analysis** of an entire process family at once
   - Reasoning about all variants collectively to determine if they meet certain properties in dimensions such as security, privacy, safety and correctness

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families
Advantages

Explicit modeling of process variation may help with:

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families.

Legend:
- \( \cdot V \) variant
- Process family

Characterizing Process Variation, Simidchieva and Osterweil

ICSE NIER Track, May 25 2011
Advantages

Explicit modeling of process variation may help with:

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families

Legend:
- **V** variant
- **Process family**

![Diagram](http://www.cs.umass.edu/~bis/)

*Characterizing Process Variation*, Simidchieva and Osterweil

ICSE NIER Track, May 25 2011
Explicit modeling of process variation may help with:

3. **Navigation** among interrelated software families
   - Identifying which pre-existing variant is most appropriate to use in specific circumstances through navigation through possibly multiple families

**Legend:**
- \( \cdot V \): variant
- \( \cdot V \): Process family

**Diagram:**
- **Functional invariance**
- **Performance variation**
- **Service Variation**

**Equations:**
- \( A \cdot V_i \)
- \( V_k \)
- \( V_j \)
- \( V_m \)
- \( V_n \)

**References:**
- Characterizing Process Variation, Simidchieva and Osterweil
- ICSE NIER Track, May 25 2011
Future Work

- How is variation rigorously and precisely defined?
- Do these dimensions afford for observed variation?
- How can families based on different variation relations be composed together safely?
- How would composition and intersection affect reasoning?
- How does process variation differ from product variation?
- What kind of tool support would make such a conceptual framework useful?
Conclusion

- Variation is inherent in real-world systems
- Being precise about different variation needs can lead to a taxonomy of different variation dimensions
- A disciplined way to model variation explicitly has the benefits of improved generation, analysis, navigation