Banff International Research Station
for Mathematical Innovation and Discovery

Advance Programme for Workshop - BX Theory & Applications Across Disciplines

Location: The seminar will be held at the TransCanada Pipelines Pavilion (TCPL) Lecture Theatre.
Breakfast, Lunch and Dinner (buffet) will be served in the Sally Borden Building.
Coffee/tea will be served outside the TCPL Lecture Theatre.

Arrival: Check in begins at 4pm on Sunday. Dinner is available that day from 5:30-7:30.
Participants are invited to meet for an informal gathering at 8pm in the 2nd floor lounge at Corbett Hall.

Talks & Demos: Participants are invited to give short "position statement" presentations (10 mins). Pls. send a title & abstract to jens@uvic.ca
We are also interested in seeing your tool demos. If you have something to demonstrate, please contact jens@uvic.ca

Programme at a glance

v.2

BX Theory & Applications **Across Disciplines**

Sunday December 1st	Monday, December 2nd	Tuesday, December 3rd	Wednesday, December 4th	Thursday, December 5th	Friday, December 6th
TIME	TCPL Lecture Theatre	TCPL Lecture Theatre	TCPL Lecture Theatre	TCPL Lecture Theatre	TCPL Lecture Theatre
	7:00-8:45 Breakfast (Sally Borden Building)	*7:00-8:45 Breakfast (Sally Borden Building)*	*7:00-8:45 Breakfast (Sally Borden Building)*	*7:00-8:45 Breakfast (Sally Borden Building)*	*7:00-8:45 Breakfast (Sally Borden Building)*
	8:45-9:00 Introduction and Welcome by BIRS Manager	**8:45-9:00** Day 1 Summary and forward planning	**8:45-9:00** Day 2 Summary and forward planning	**8:45-9:00** Day 3 Summary and forward planning	**8:45-9:00** Seminar Summary
	9:00-9:15 Introduction and overview by Organizers	**9:00-10:15** Challenge problems and benchmarks - proposals	**9:00-10:15** Position statements (Session 1) Theme: BX in Programming Languages *Chair: Richard Paige*	**9:00-10:15** Position statements (Session 3) Theme: BX in Software Engineering *Chair: Jens Weber*	**9:00-10:15** Book planning, TOC and chapter proposals
	9:15-10:15 Minitutorial: BX in PL (Prog. Languages) by Janis Voigtlander				
	10:15-10:45 Break	*10:15-10:45 Break*	*10:15-10:45 Break*	*10:15-10:45 Break*	*10:15-10:30 Break*
	10:45-11:45 Minitutorial: BX in SE (Software Engineering) by Zinovy Diskin	**10:45-11:00** Form breakout groups: *How to structure & curate a challenge?*	**10:45-12:00** Position statements (Session 2) Theme: BX in Graph Transformations *Chair: Andy Schürr*	**10:45-12:00** Position statements (Session 4) Theme: BX Applications & Systems *Chair: James Terwilliger*	**10:30-11:30** Discussion of possible joint initiatives & Wrap up
		11:00-12:00 Group work (accross disciplines)			Check out of room by 12 pm
	11:45-1:00 Lunch (Sally Borden Building)	*12:00-1:30 Lunch (Sally Borden Building)*	*12:00-1:30 Lunch (Sally Borden Building)*	*12:00-1:30 Lunch (Sally Borden Building)*	*11:30-1:30 Lunch (Sally Borden Building)*
	*1:00-2:00 **Guided Tour** of The Banff Centre; meet in the 2nd floor lounge, Corbett Hall* *2:00 **Group photo**, meet in foyer of TCPL, bring jacket*	**1:30-2:15** Report and discussion **2:15-3:00** Selection of THE	*Social Event / Excursion to Lake Louise*	**1:30-2:00** Working group proposals	
	2:15-3:15 Minitutorial: BX in GT (Graph Transformation) by Frank Hermann	BIRS-BX Challenge / Benchmark(s)™ **3:00-3:15** Form groups: How would you attack the problem in your discipline?		**2:00-3:15** Group Work	*Participants are welcome to use BIRS facilities (BIRS Coffee Lounge, TCPL and Reading Room) **until 3 pm***
	3:15-3:45 Break	*3:15-3:45 Break*		*3:15-3:45 Break*	
4:00 Check-in begins (Front Desk – Prof. Development Centre - open 24 hours)	**3:45-4:45** Minitutorial: BX in DB (Databases) by James Terwilliger	**3:15-4:15** Group work (accross disciplines)		**3:45-4:45** WG Reports	
	4:45-5:15 Challenge Problems and Benchmarks - *Motivation Session* (Perdita Stevens, Jeremy Gibbons)	**4:15-5:15** Reports and Discussion		**4:45-5:15** Discussion of possible edited volume on BX	
5:30-7:30 Buffet Dinner Sally Borden Building	*5:45-7:00 Dinner Sally Borden Building*	*5:45-7:00 Dinner Sally Borden Building*	*5:45-7:00 Dinner Sally Borden Building*	*5:45-7:00 Dinner Sally Borden Building*	
	7:00-8:00 Optional discussions on challenge problem proposals	**7:00-8:00** Optional time to focus position statements on challenge problem	**7:00-8:00** Tool Demos (Session 1)	**7:00-8:00** Tool Demos (Session 2)	
8:00 Informal gathering in 2nd floor lounge, Corbett Hall					

Position Statements Session 1 – Details

Theme: BX in Programming Languages

Wednesday, 9-10:15

Jeremy Gibbons: *"Incremental Updates and Non-free Datatypes"*

A BX may be sufficiently well behaved to support incremental updates - that is, for an edit affecting a small part of the view to result in a correspondingly small update to the source. I have some idea of how to capture this notion for BXs on tree-like datatypes, ie those providing unique decompositions of terms into subterms. I would like to understand this better on non-free datatypes, ie those without such unique decompositions. Lists provide a simple example: asymmetric "cons" lists are a tree-like datatype, but this doesn't lead to a helpful class of incremental updates; symmetric "append" lists are more helpful, but don't fit the theory. More importantly, graphs aren't a tree-like datatype, so how to capture incremental updates on graphs? I think in general we need a better understanding of algebraic approaches to non-free datatypes. I conjecture that Abbott's "containers" approach and Joyal's combinatorial species will be relevant.

James KcKinna: "Type Theory for Least Change"

Towards our long-term goal of understanding, and explaining, ideas of "least change" in the context of bx, we have been looking at trying to unify the various (extensional) approaches to bx in terms of additional (intensional) structure expressible in (dependent) type theory.

Michael Johnson and Bob Rosebrugh: *"Mathematical Foundations for BX: Symmetric lenses and spans"*

Our position is that, like other areas of Software Engineering, BX needs a mathematical foundation both to systematize extant work and to help guide development. One example we are currently pursuing follows.

There are several kinds of asymmetric lenses depending on what structure is assumed for the model domain and what is required of the put operation. Several related kinds of symmetric lenses address problems of model synchronization. The relationship between symmetric and asymmetric lenses needs to be made precise. We are developing a theory which unifies the multiplicity of lens structures, revealing new lens structures and clarifying lens equivalence.

Meng Wang: Making Semantic Bidirectionalization More Applicable

Bidirectionalization is a program transformation technique turning uni-directional programs into bidirectional ones. Among others, a technique termed as semantic bidirectionalization pioneered by Janis Voiglander stands out in term of user-friendliness. The unidirectional program can be written using arbitrary language constructs, as long as the function is polymorphic. This is convenient, in the sense that the programmer is not restricted to using a particular syntax; but it does require the transformation to be polymorphic.

Notable progress has been made since, in particular on the relaxation of the polymorphism restriction. We are now able to apply similar techniques to a wider range of polymorphic functions, and in some cases even to monomorphic ones.

Position Statements Session 2 – Details

Theme: BX in Graph Transformations

Wednesday, 10:45-12:00

Frank Hermann: "*Correctness and Completeness for Incremental Model Synchronisation Based on TGGs*"

Triple graph grammars (TGGs) have been applied success-fully for specifying and analysing bidirectional model transformations. In cases where model modifications may occur in multiple domains and at any time, bidirectional transformations need to provide a mechanism to achieve concurrent model synchronisation. Changes in one domain are propagated to a related domain and conicts between different model updates need to be solved. In order to ensure correctness and completeness, a formal approach to concurrent model synchronisation has been presented for arbitrary TGGs. Several tools for TGGs provide mechanisms for incremental model synchronisation requiring some restrictions on the used TGGs and they do not concern the concurrent case, i.e., updates cannot occur simultaneously on multiple domains. From a formal point of view, we think that the following future challenges are highly relevant.

1. To provide a generalisation of existing formalisations concerning the notion of incrementality of model synchronisation using an adequate notion of least change.
2. To relate and extend the formal notions and the existing implementations for model synchronisation in order to ensure correctness, completeness and incrementality of concurrent synchronisation within the tools.

Holger Giese: "*The Implications of Optimality Results for an Incremental Model Synchronization Algorithm with TGGs*"

We were lately able to develop an incremental model synchronization algorithm that is optimal for a large class of model transformation problems, which have been identified based on a thorough analysis of the employed meta models and model transformation specifications. However, the obtained results raise a number of interesting research questions: (1) Can similar optimal results also be optioned for a larger class of model transformation problems and in particular more expressive model synchronization specifications? (2) Are these results relevant also for synchronization problems in other domains? Consequently, we suggest to study in more depth the characteristics of the synchronization problems in the different domains to better understand where the communities can benefit from each other and where probably different directions for appropriate solution have to be studied.

Fernando Orejas: "*Incrementality in Model Transformation with Triple Graph Grammars*"

Abstract forthcoming

Soichiro Hidaka: "*Compositional development framework for bidirectional model transformations based on structural recursion on graphs*"

In this position statement, we introduce our integrated framework called GRoundTram (Graph Roundtrip Transformation for Models), a tool supporting for systematic development of Bidirectional model transformation. It is designed and implemented for compositional development of well-behaved and efficient bidirectional model transformations. GRoundTram is built upon a well-founded bidirectional framework and is equipped with a user-friendly textual language for coding bidirectional model transformations, a novel tool for validating both models and transformations, an optimization mechanism for improving efficiency, and a powerful debugging environment for testing bidirectional behavior. We also introduce our recent usage of GRoundTram, including co-evolution of model and code, collaborative development of bio-models, and feedback of formal verification results, as well as our future directions including interoperability and enhancement of graph models such as graphs with ordered branches.

Position Statements Session 3 – Details

Theme: BX in Software Engineering

Thursday, 9:00-10:15

Albert Zündorf: "*Doing Model to Model transformations inplace with an intermediate generic graph*"

For the Transformation Tool Contest 2013 one challenge was to do a simple model to model transformation where some attributes and classes were renamed and relocated. In SDMLib I did this using a reflective mechanism to tranform the input model into a generic graph (where the node type is an attribute and the node attributes are just a set of tag value pairs). In this generic graph it was possible to rename e.g. attributes and to change the type of an object. Then SDMLib used reflictive mechanisms again to translate the result into the target model. With the help of the intermediate generic graph, SDMLib uses simple inplace model transformations to do the model to model transformation. For such simple model transformations it is quite clear and easy how to reverse them. Thus, this migth be a simple way to achieve bidirectional M2M transformations.

Yingfei Xiong: BX from Synchronous Grammars

Typical ways of constructing BXs is to use functional combinators or logic constraints. In this position statement, we propose to a novel method to construct BX: using

synchronous grammars and data mergers. Synchronous grammars are proposed by natural language processing community in 1960s and are designed for specifying the recursive relationship between two languages. In this in-progress work, we conjecture that, by equipping with proper data mergers, we could express many BX relations using synchronous grammars, and the expressiveness is no weaker than typical combinator-based languages such as Boomerang. There are several benefits of using this approach: 1) existing work on parser generation could be integrated, providing a solid theoretical foundation and saving development cost; 2) developers could rely on their existing knowledge of grammars, flattening the learning curve; 3) complex cases, such as matching the parentheses, could be easily expressed in this approach.

Arend Rensink: Semantics-Preserving Transformation: An Impossible Dream?

Not many researchers or practitioners would deny that a model transformation should preserve the meaning of model being transformed. Problems start to arise when attempting to define more precisely what this "meaning" is that needs to be preserved. One possible definition is "operational semantics": under this interpretation, I will argue that we are currently lacking in methods to prove preservation.

Zhenjiang Hu: Validation of Bidirectional Transformation

Abstract forthcoming

Position Statements Session 4 – Details

Theme: BX Applications and Systems

Thursday, 10:45-12:00

James Cheney: "*Bidirectionality, traceability and provenance*"

A number of bidirectional transformation techniques employ descriptions (traceability links, alignment, witness structures) of the relationships between objects of interest. The meaning of these traces is not always part of the functional description of what a transformation should do, but seems to play an important role in defining what it does do.

A number of programming and computation models can produce descriptions (provenance, lineage, traces) of the relationships between the input and output of a computation. The meaning of these traces is not always characterized in terms of formal requirements on the behavior of the overall system, but seems to meet informal requirements, for e.g. reproducible scientific computation.

Is traceability information a form of provenance (and vice versa)? Can we use ideas from provenance to improve understanding of witness structures (or vice versa)?

Tom Maibaum: Why formal approaches are not just necessary but good for your health!

Abstract forthcoming

Mathieu Beine, Nicolas Hames, Jens Weber: "Implementing BX for database schema / application co-evolution - performance comparison and alternatives"

Bidirectional transformations (bx) play an important role in database schema / application co-evolution. In earlier work, Terwilliger introduced the theoretical concept of *channels* as a transformation-based mechanism to de-couple the "virtual database" used by the application code from the actual representation of the data maintained within the DBMS. In this presentation, we report on experiences implementing such channels in practice in the context of complex real-world applications. We focus on channels implementing pivot/unpivot transformations. We present different implementation alternatives for such channels and discuss their performance characteristics. Finally, we present a transformational tool to generate these channels.

Frank Hermann: "Application of TGGs for translating satellite control procedures"

Abstract forthcoming

Tool Demos

Tool Session 1

Wednesday, 7:00-8:00

Soichiro Hidaka: "*GRoundTram: An Integrated Framework for Developing Well-Behaved Bidirectional Model Transformations*"

We introduce our integrated framework called GRoundTram (Graph Roundtrip Transformation for Models), a tool supporting for systematic development of Bidirectional model transformation. It is designed and implemented for compositional development of well-behaved and efficient bidirectional model transformations. GRoundTram is built upon a well-founded bidirectional framework and is equipped with a user-friendly textual language for coding bidirectional model transformations, a novel tool for validating both models and transformations, an optimization mechanism for improving efficiency, and a powerful debugging environment for testing bidirectional behavior. In this demonstration we also show our recent application scenario on collaborative development of bio-models, and feedback of formal verification results.

Albert Zuendorf: "*SDMLib*"

I would like to demonstrate SDMLib a new light weight model transformation approach. Especially, I would like to show how SDMLib supports model to text transformations aka code generation and how SDMLib text templates may also be used for parsing i.e. for text to model transformations.

Alcino Cunha: "*Model Repair and Transformation with Echo*"

Models are paramount in model-driven engineering. In a software project many models may coexist, capturing different views of the system or different levels of abstraction. A key and arduous task in this development method is to keep all such models consistent, both with their meta-models (and the respective constraints) and among themselves. This demo will present Echo, a tool that aims at simplifying this task by automating inconsistency detection and repair using a solver based engine. Consistency between different models can be specified by bidirectional model transformations, and is guaranteed to be recovered by minimal updates on the inconsistent models. The tool is freely available as an Eclipse plugin, developed on top of the popular EMF framework, and supports constraints and transformations specified in the OMG standard languages OCL and QVT-R, respectively. More information available in the tool webpage at http://haslab.github.io/echo/.

Tool Session 2

Thursday, 7:00-8:00

Anthony Anjorin: "*eMOFLON*"

The demo of eMoflon (www.emoflon.org), an actively developed Triple Graph Grammar tool, will focus on the following points:

- Giving a brief overview of a new industrial application in the domain of manufacturing engineering
- Giving a feel for the tool and presenting some of the latest features (e.g., modularity concepts) implemented to deal with a real-world specification of considerable size
- Demonstrating how we specify "deltas", which can be propagated incrementally by our tool (i.e., using TGGs for incremental model synchronization)

Arend Rensink: "*The GROOVE Tool*"

Abstract forthcoming

Martin Gogolla: "*Modeling Transformations with USE*"

We assume that a modeling language can be described by two metamodels, one for its syntax and one its semantics where usually the semantics metamodel depends on the syntax metamodel. When two languages, say the language North and the language South, are to be transformed into each other, we describe this by a direction-neutral transformation metamodel establishing connections to both languages incorporating their syntax and semantics. Properties of the transformation like equivalence or embedding must be formulated as constraints in the transformation metamodel. We explain how the UML and OCL tool USE can be employed for testing such transformations.

Frank Hermann: "*HenshinTGG*"

Abstract forthcoming