A systematic approach to constructing families of incremental topology control algorithms using graph transformation

Presentation at ICGT 2018
Toulouse, France, 2018-06-25

Roland Kluge
roland.kluge@es.tu-darmstadt.de
with Michael Stein, Gergely Varró, Andy Schürr, Matthias Hollick, Max Mühlhäuser
Supported by the Cooperative Research Center 1053 "MAKI" – https://tiny.cc/MAKI

Example: da_sense – A hybrid sensor network for Smart Cities ("Digitalstadt Darmstadt") (I)

An Urban Management Platform based on heterogeneous sensor networks
- Traffic: static at traffic lights
- Temperature: mobile on trams
- Noise: mobile via Android app

http://www.da-sense.de/
Example: da_sense – A hybrid sensor network for Smart Cities ("Digitalstadt Darmstadt") (II)

An Urban Management Platform based on heterogeneous sensor networks

- Traffic: static at traffic lights
- Temperature: mobile on trams
- Noise: mobile via Android app

Image Sources:
- Tram: "Die Transparente Stadt" https://www.youtube.com/watch?v=3imyRNylZKo
- Android screenshots and map: http://www.da-sense.de/
Graph-based topology model for Wireless Sensor Networks

Transmission range of $n_1$

Wireless node $n_1$

Wireless link $e_{12}$ with link weight $w(e_{12})$

TelosB sensor node
48kB ROM,
10kB RAM,
2xAA battery

Topology control sparsens topologies to improve non-functional property
**Topology control by example: kTC algorithm**

**kTC rule:**
"Inactivate a link if (and only if) it is
(i) the weight-maximal link in a triangle and
(ii) at least $k$-times longer than the weight-minimal link in the triangle."

---

**Input topology**

**Output topology**

- **Active**
- **Inactive**

- Reduced transmission range of $n_1$

$k=1.2$
The curse of low abstraction in traditional communication system development

Theorem V.1. \( G_{\text{KTC}} \subseteq G_{\text{GG}} \), or equivalently, the diametrical circle of any two nodes \( u, v \in G_{\text{KTC}} \) is empty.

Proof: We will show that \( (u, v) \notin G_{\text{GG}} \) implies that \( (u, v) \notin G_{\text{KTC}} \). Pick \( (u, v) \in G - G_{\text{GG}} \). Then there must exist a \( w \in G \) such that \( w \) lies inside the diametrical circle of \( u \) and \( v \). By the assumption of the UDG, \( (u, v) \in G \) implies that \( (u, v) \in G_{\text{KTC}} \). Without loss of generality, assume \( u \) and \( v \) are oriented horizontally. The maximum value of \( \min(d(u, w), d(v, w)) \) is then attained on the top or bottom of the diametrical circle where \( d(u, w) = d(v, w) \). The maximum ratio of \( d(u, w) \) to \( d(v, w) \) is thus \( \sqrt{2} \). Since KTC is only defined for \( k < \sqrt{2} \), the edge \( (u, v) \) is also discarded by KTC.

Corollary V.2. \( G_{\text{KTC}} \) is planar.

Proof: The Gabriel Graph \( G_{\text{GG}} \) is planar.

Theorem V.3. \( G_{\text{KTC}} = G_{\text{KTC}} \) whenever \( k = 1 \).

Proof: For clarity, we ignore the tie-breaking case, where both XTC and KTC discard the same edge based on IDs. In XTC an edge \( (u, v) \) is removed if there is a node \( w \) with \( d(u, w) < d(u, v) \) and \( d(v, w) < d(u, v) \). Nodes \( u, v, \) and \( w \) form a triangle where \( (u, v) \) is the longest edge. When \( k = 1 \), KTC exactly lengthens the longest edge.

Corollary V.4. \( G_{\text{XTC}} \subseteq G_{\text{KTC}} \).

Proof: Increasing \( k \) only adds edges.

Image Sources: FlockLab https://www.flocklab.ethz.ch/wiki/chrome/site/wiki_public/observer/outdoor_1.jpg

Kluge et al. | Real-Time Systems Lab | Prof. Dr. Andy Schürr | A systematic approach to constructing families of incremental topology control algorithms using graph transformation
Goal: Support the correct-by-construction development of topology control algorithms

How to specify consistency (declaratively)?
- Formalization, proofs, ...

How to operationalize consistency preservation?
- Localization

(i) Specification:
- Graph theory, ...

(ii) Simulation:
- C, Java, ...

(iii) Testbed:
- C, C++, ...

Limitations ✓
Constraints ✓

How to ensure transparency/traceability?

Model

Code

Iteration

Extensible code gen.

Level of abstraction
MAKI for a better Future Internet
Multi-Mechanismen-Adaption für das Künftige Internet

C: Communication systems
(concrete self-adaptive systems)

B: Adaptation mechanisms
(reusable components for adaptive systems)

A: Construction methods
(models, design patterns, languages)

- Network topologies and topology adaptation
- Specification languages
- Software engineering

P2P Streaming
Mesh Hybrid Tree

C/S Streaming
DASH HLS ...

1. filter()
2. max()
3. join(
4. match(TP, T1, self <- e0 -> n1,
5. n1 <- e1 -> n2),
6. match(TP, T3, self <- e2 -> n3)),
7. e0.weight),
8. count(
9. match(TP, T2, self - e3 -> n4)) = 0
10. execute every match:
11. at(self, TP, T2) add neighbor(n1)

https://www.maki.tu-darmstadt.de/
Unstructured consistency specification

Unstructured formulation

kTC: "Inactivate a link if (and only if) it is
(i) the weight-maximal link in a triangle and
(ii) at least k-times longer than the weight-minimal link in the triangle."

Structured formulation

\[ \varphi(e_{12}, e_{13}, e_{32}) = \]
\[ \wedge w(e_{12}) > \max(w(e_{13}), w(e_{23})) \wedge w(e_{12}) > k \cdot \min(w(e_{13}), w(e_{23})) \]

Problem 1: Implicit, unstructured, or informal specification of constraints.
Topography control algorithms form families

— Family: common structural pattern
— Algorithm: refinement based on attribute constraints

Problem 2: Insufficient usage of relationships among topology control algorithms
TC Algorithm families are Dynamic Software Product Lines

Advantages: Reuse in of predicates, spec. of topology control reconfiguration

\[
\varphi(e_{12}, e_{13}, e_{32}) = \\
w(e_{12}) > \max(w(e_{13}), w(e_{23})) \wedge \\
w(e_{12}) > k \cdot \min(w(e_{13}), w(e_{23}))
\]
Graph constraints for specifying local consistency properties

"Each inactive link should be part of a triangle for which $\varphi$ holds"

"No active link should be part of a triangle for which $\varphi$ holds"

Advantages: expressiveness, formal + domain-specific + operationalizeable
Example: Fulfilled and violated constraints

Example:

\[ \varphi(e_{ab}, e_{ac}, e_{cb}) \]

Fulfilled constraints:

\[ \varphi(e_{ab}, e_{ac}, e_{cb}) \]

Violated constraints:

\[ \varphi(e_{ab}, e_{ac}, e_{cb}) \]
Enforcing and preserving consistency

Consistency enforcement: Recover from any inconsistent state

Consistency preservation: Weak and strong consistency
Recap on consistency preservation

Global consistency properties

implies

Local consistency properties (declarative)

preserves

Topology control algorithm specification (operationalized)

Generic
Algorithm-specific

→ e.g., connectivity
→ e.g., "no ϕ-triangles"

Global consistency properties

implies

Local consistency properties (declarative)

preserves

Topology control algorithm specification (operationalized)

?
Specifying algorithmic implementation using Story-Driven Modeling [FNT+98]

1. Start
2. Find unmarked link
3. Try activating $e_{12}$
4a. If successful, continue.
4b. If unsuccessful, try inactivating $e_{12}$
5. Continue in any case.
6. Terminate if no more unmarked links exist.

Spec. Sim. Test.


Kluge et al. | Real-Time Systems Lab | Prof. Dr. Andy Schürr | A systematic approach to constructing families of incremental topology control algorithms using graph transformation
Rule applications may violate consistency!

Apply

\[ R_a(e_{12} : \text{Link}) \]

LHS \[ e_{12} \]

RHS \[ e_{12} \]

at link \( e_{13} \)

\[ \varphi(e_{ab}, e_{ac}, e_{cb}) \]
Ensuring inductive consistency preservation

Pairwise refinement

\[ R_x \]  
\[ C_y \]

Repeat for all \((R_x, C_y)\) pairs

Example: Refining \(R_a\) based on \(C_a\)

Advantages: mechanical algorithm, existing tool support

\[ \text{Spec.} \quad \text{Sim.} \quad \text{Test.} \]


Refined algorithm specification

**Advantages:** inductive invariant, strong consistency on termination

**Required:** proof of termination
Topologies are never stable: Context events and dynamic topology control

Context events (CEs) reflect environmental influences

Example: Link removal rule

\[
\begin{array}{c|c}
\text{LHS} & \text{RHS} \\
1 & 2 \\
\end{array}
\]

- Remove \( e_{12} \)
- Remove \( e_{21} \)

Obstacle

TC

R

-e(\( e_{12} \) : Link)
Context event handlers: Anticipating consistency violation

\[ R_e(e_{12} : \text{Link}) \]

\[
\begin{array}{c|c}
\text{LHS} & \text{RHS} \\
\hline
1 & 2 \\
\end{array}
\]

\[ R_e(e_{12} : \text{Link}) \]

\[
\begin{array}{c|c}
\text{LHS} & \text{RHS} \\
\hline
1 & 2 \\
\end{array}
\]

\[ \text{PAC}_{e,i,2} \]

\[ \text{PAC}_{e,i,6} \]

\[ \text{Spec.} \quad \text{Sim.} \quad \text{Test.} \]
Consistency preservation by context event handling

— Context event handler for $R_e$

handle-$R_e(e_{12}: \text{Link})$

— Achieved consistency preservation

Pending context event

Remove $e_{12}$

Inactivate $e_{13}$
Recap on specification phase

- **Global consistency properties**
  - Generic
  - Algorithm-specific
  - e.g., connectivity
  - e.g., "no \( \varphi \)-triangles"

- **Local consistency properties** (declarative)

- **Topological control algorithm specification** (operationalized)

- **[HW95, DV14]**
Tool support for simulation and testbed

Simulation

Java API

C Code

Testbed

Spec. Sim. Test.


Ongoing Work: From centralized to distributed topology control algorithm specifications

Centralized-global perspective of TC:
TC: sequential
Topology: global, consistent view

TC: distributed
Topology: local consistent view

Distributed-local perspective of TC:
TC: distributed
Topology: local, inconsistent + monitoring
Computation model

Goal: Characterize concurrent execution + identify potential problems

- **Atomic actions**: rule applications and synchronized view of topology
- **Interleaved execution**: no two events at same point in time
- **Vertex-centric**: A node has 1 process and 1 local-view model

**Centralized execution**

- TC
- Antic.
- CE
- TC
- Antic.
- CE
- Antic.
- CE
- TC

**Distributed execution**

1. T
2. T
3. A
4. A
5. A
6. A
7. A
8. A
9. A
10. T

**Global control flow**

**Per-node control flow ("happened-before")**
Example: Non-termination of context handling

Pending context event

Liveness problem due to Mutual dependency (necessary)
Return to same state (sufficient)
Take-Home Messages

**Goal:** Overcome curse of low abstraction in dev. of topology adaptations

**Result:** Correct-by-construction development methodology TC algorithm families

**Ongoing work:** Toward distributed-local topology control specifications

(i) **Specification:** Graph theory, ...
(ii) **Simulation:** C++, Java, ...
(iii) **Testbed:** C, Assembler-like, ...

Summary:
- Formalization, proofs, ...
- Localization
- Extensible code gen.
- Feedback & refinement

Supplementary material on GitHub
References

- **eMoflon/cMoflon logos**: Work of the eMoflon developer team. Subject to Fair Use conditions.
- **Peerfact logo**: Multimedia Communications Lab (KOM), TU Darmstadt, [https://www.kom.tu-darmstadt.de](https://www.kom.tu-darmstadt.de)
- **Henshin logo**: [https://www.eclipse.org/henshin/](https://www.eclipse.org/henshin/)
- **EMF logo**: [https://www.eclipse.org/modeling/emf/](https://www.eclipse.org/modeling/emf/)
- **Android logo**: [https://developer.android.com/distribute/marketing-tools/brand-guidelines](https://developer.android.com/distribute/marketing-tools/brand-guidelines) "The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License."
- **TelosB Sensor node**: Wiki Commons, CC-BY-3.0, by Jbasic, [https://commons.wikimedia.org/wiki/File:TelosB.jpg](https://commons.wikimedia.org/wiki/File:TelosB.jpg)
- **Writing Hand**: CC-BY-3.0 US, Rediffusion, [https://thenounproject.com/search/?q=hand%20drawing&i=29383](https://thenounproject.com/search/?q=hand%20drawing&i=29383)
- **Gears**: CC-BY-3.0 US, Pedro Santos [https://thenounproject.com/search/?q=gear&i=1030299](https://thenounproject.com/search/?q=gear&i=1030299)
- **GitHub Logo**: CC-BY-3.0, Dave Gandy, [http://www.flaticon.com/free-icon/github-logo_25231](http://www.flaticon.com/free-icon/github-logo_25231)