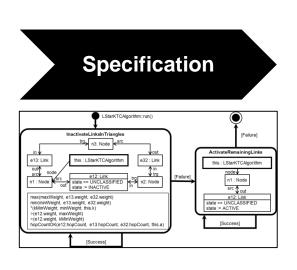
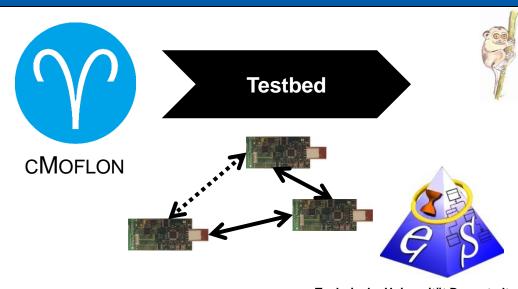
cMoflon: Model-Driven Generation of Embedded C Code for Wireless Sensor Networks





Conference Talk at ECMFA 2017 2017-07-19





Roland Kluge

roland.kluge@es.tu-darmstadt.de

Joint work with Michael Stein, David Giessing, Andy Schürr, Max Mühlhäuser

Supported by the Cooperative Research Center 1053
"Multi-Mechanism Adaptation for the Future Internet" (MAKI) – https://tiny.cc/MAKI

Technische Universität Darmstadt Fachgebiet Echtzeitsysteme – Real-Time Systems Lab

Prof. Dr. rer. nat. Andy Schürr

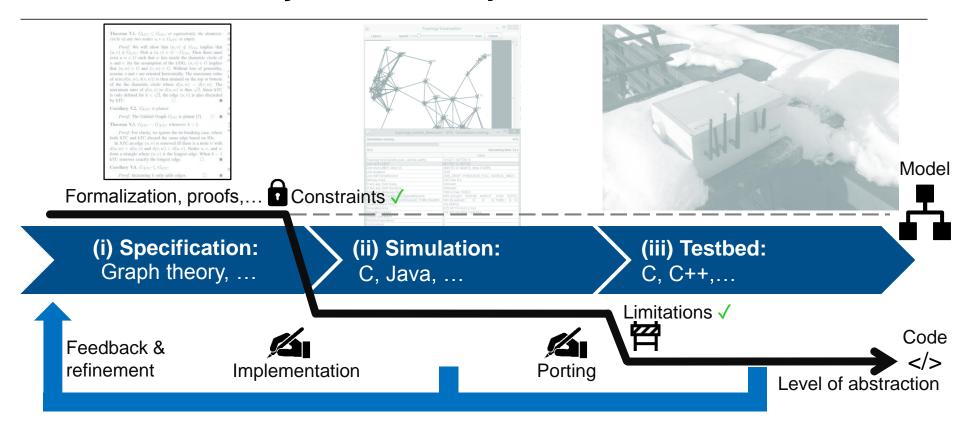
Dept. of Electrical Engineering and Information Technology

Dept. of Computer Science (adjunct Professor)

www.es.tu-darmstadt.de

The curse of low abstraction in traditional communication system development



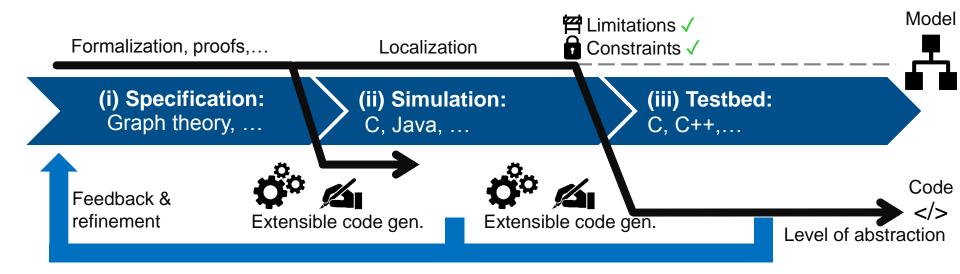




https://www.flocklab.ethz.ch /wiki/chrome/site/wiki_public /observer/outdoor_1.jpg

Leverage model-driven engineering principles!

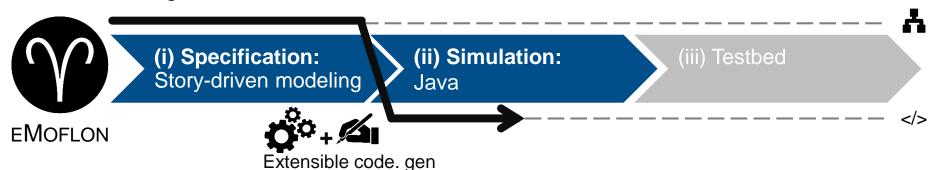


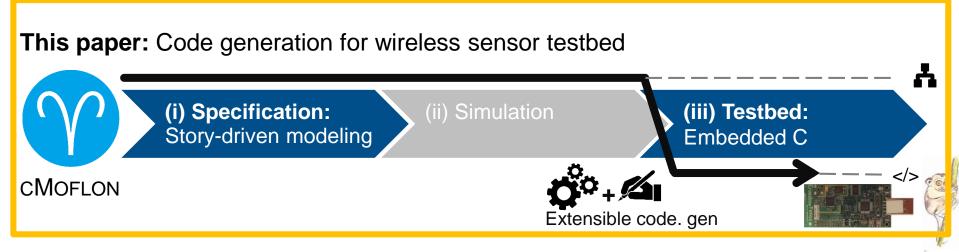


Contribution



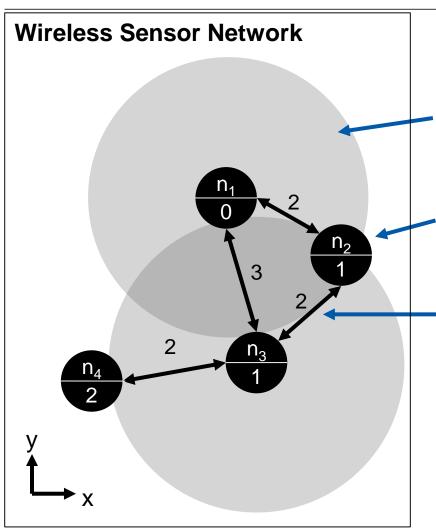
Previous: Integration with wireless network simulator





Wireless Sensor Networks





Transmission range

Wireless node n₂ + hop count h₁(n₂) to n₁

Wireless link e₃₄
+ link weight w(e₃₄)
(e.g. distance
approx. via RSSI)

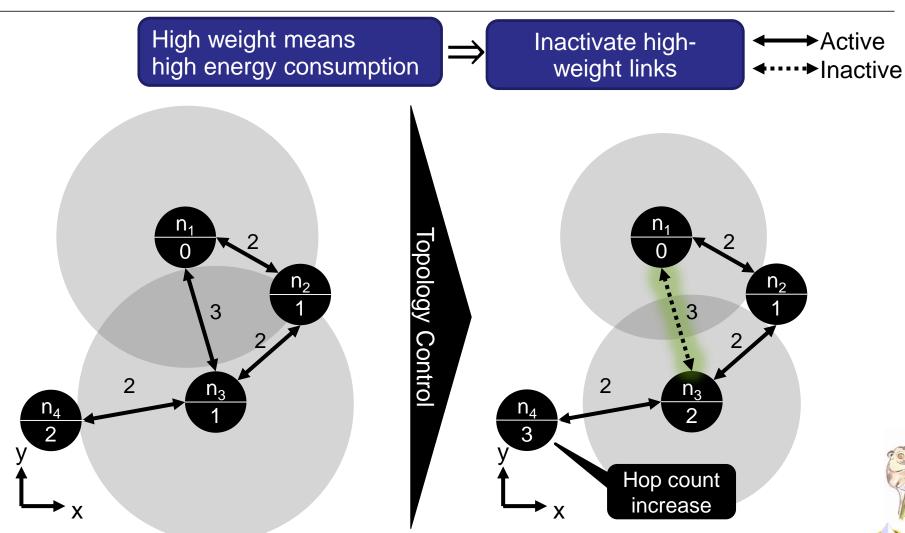


TelosB sensor node [Polastre05] (48kB ROM, 10kB RAM)

[Polastre05] Polastre et al.: "Telos: enabling ultra-low power wireless research," IPSN 2005

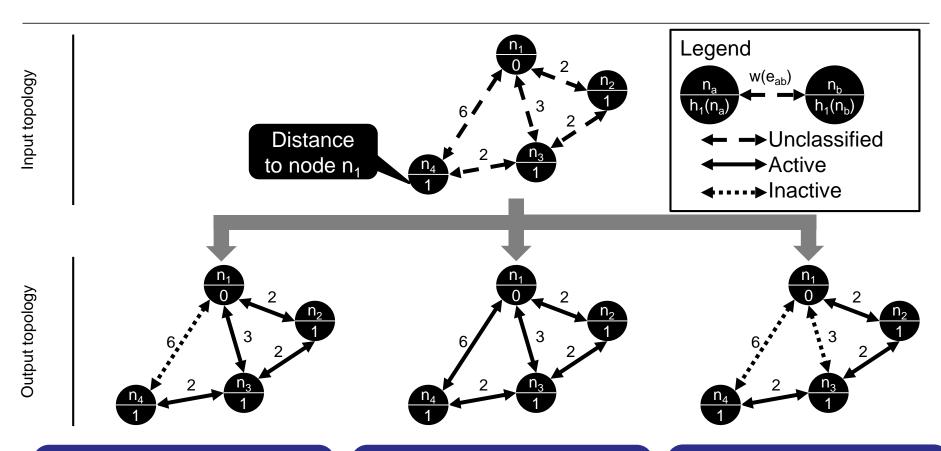
Topology Control





Running examples: kTC, I*-kTC, and LMST





kTC[Schweizer16](k=2)
Inactivate the weightmaximal link in each triangle

I*-kTC[Stein16] (k=2,a=1.5) kTC + bound increase of routing path length LMST_[Li05]
Activate links on local minimum spanning tree

Challenges while building CMOFLON



- Ensure applicability
- - **Problem:** Unrealistic to build a "one-fits-all" solution
 - **Approach**: Representative algorithms and extensibility
- Foster rapid prototyping



- Problem: Porting to testbed incures high manual effort
- **Approach**: Automation + extension points
- Respect resource limitations



- **Problem:** Resource constraints (e.g., Telos-B: 48kB ROM, 10kB RAM)
- Approach: Use ToCoCo framework for Contiki OS





SPECIFICATION

(i) Specification

(ii) Simulation

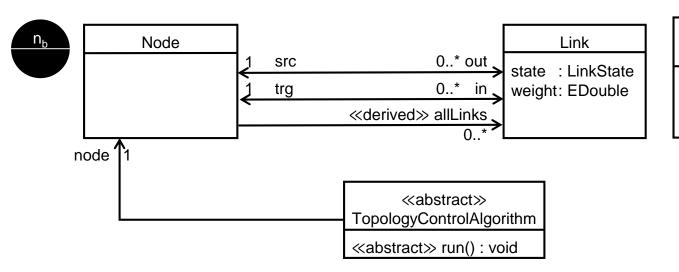
(iii) Testbed



Topology Control metamodel







≪enumeration≫ LinkState

Active : LinkState Inactive : LinkState Unclassified : LinkState

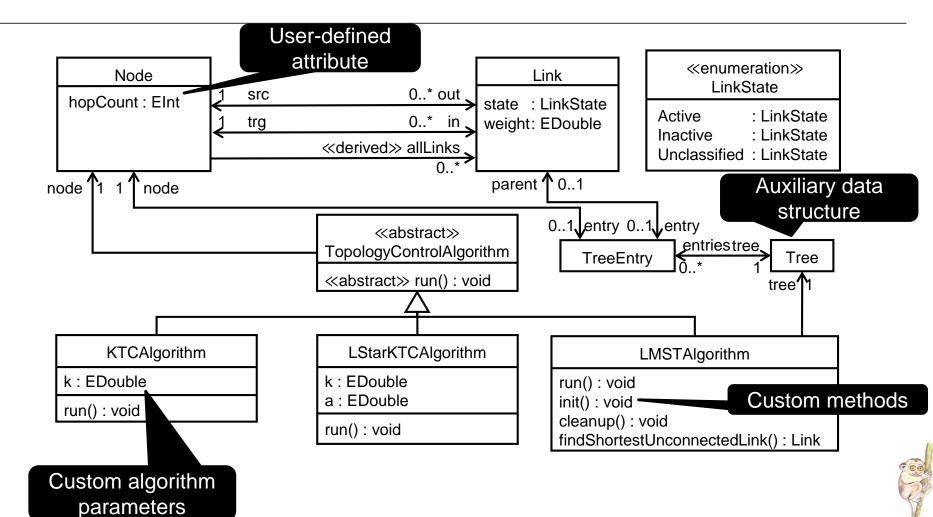




Topology Control Metamodel



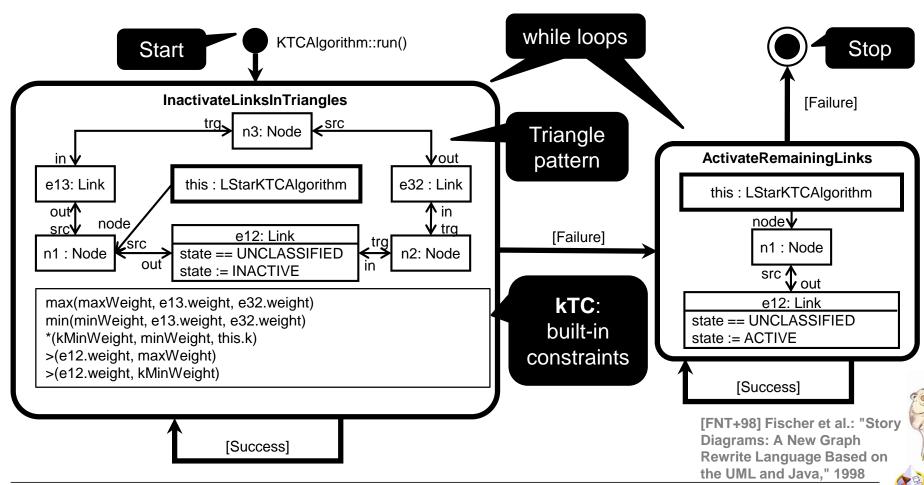




kTC story diagram

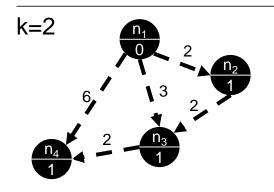




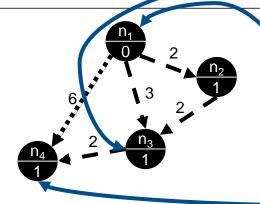




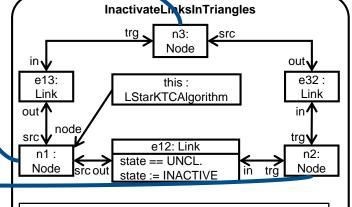
LStarKTCAlgorithm::run()



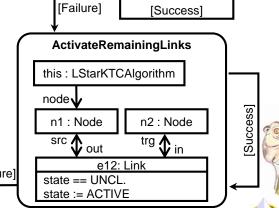
Initial



After loop "InactivateLinksInTriangle"

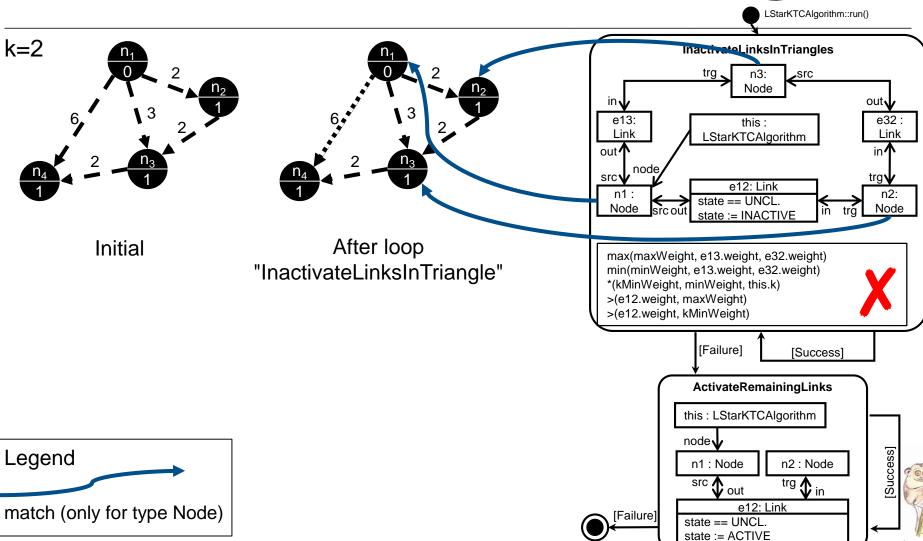


max(maxWeight, e13.weight, e32.weight) min(minWeight, e13.weight, e32.weight) *(kMinWeight, minWeight, this.k) >(e12.weight, maxWeight) >(e12.weight, kMinWeight)

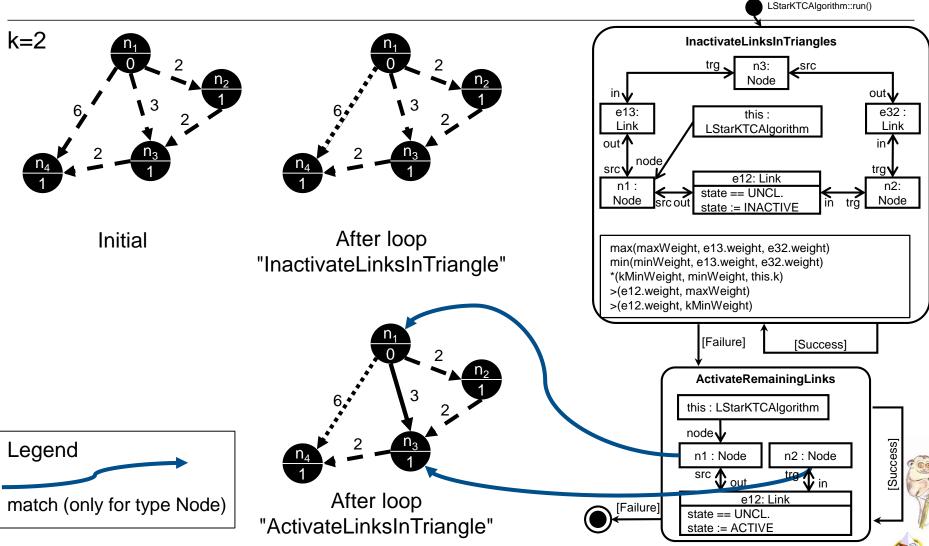


Legend match (only for type Node)

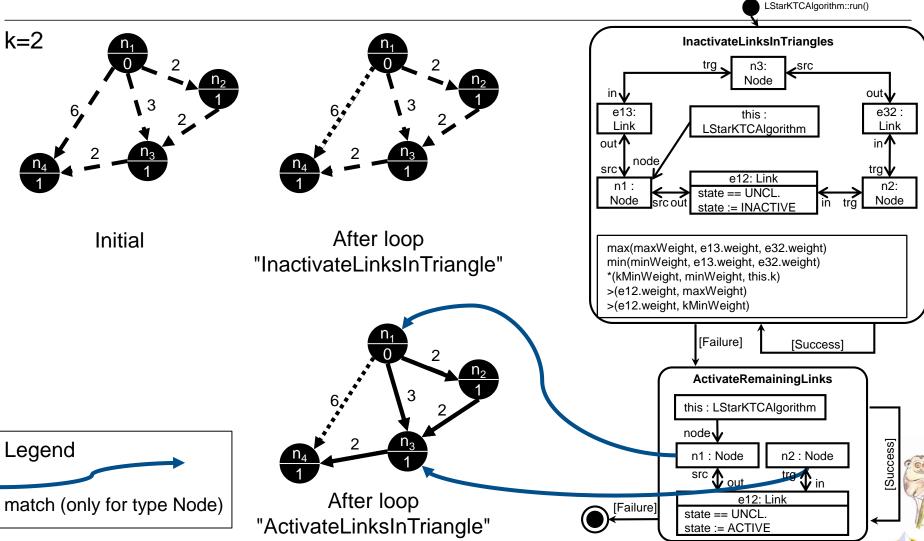








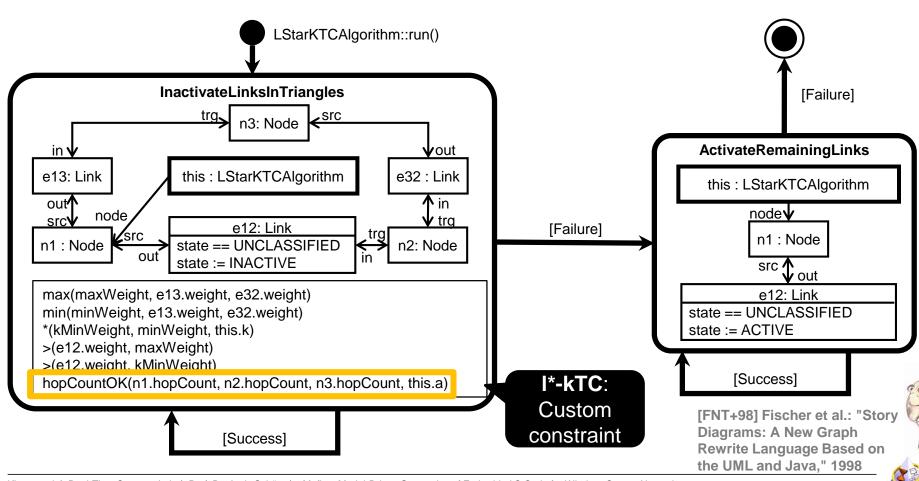




I*-kTC story diagram









CODE GENERATION FOR TESTBED EVALUATION

(i) Specification

(ii) Simulation

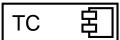
(iii) Testbed



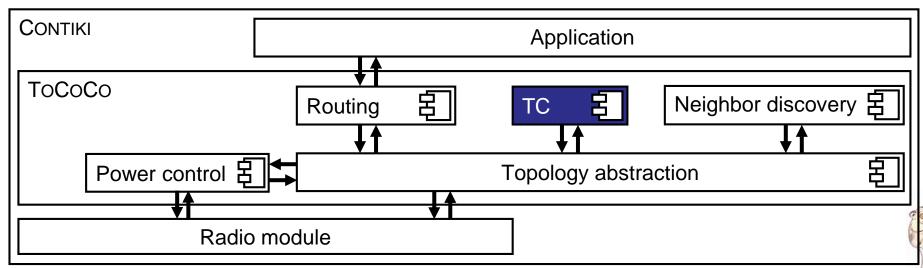
CONTIKI and ToCoCo as target platform



- Contiki OS: widely used in the WSN community
- ToCoCo Topology Control framework for Contiki: rapid prototyping and evaluation of Topology Control algorithms
- —cMoflon creates a | ⊤



per subclass of



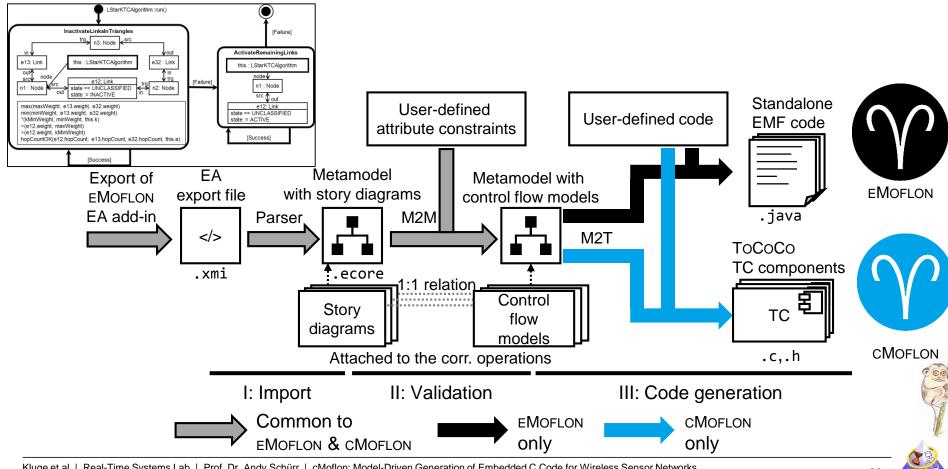
[Stein16] Stein et al.: "Control in Wireless Sensor Networks: What Blocks the Breakthrough?," In: LCN 2016 Source code on GitHub: https://github.com/steinmic/ToCoCo



A lean variant of EMOFLON



- Extensive reuse: Modeling frontend, preprocessing
- Exchanged code generation phase (ca. 3400 LOC, 1% of EMOFLON)



Example: Control flow code

Array of void*



```
Auxiliary type: Developer provides
 Memory management (I):
 free after use
                                            mapping (i.e., C struct)
                                                                                             LStarKTCAlgorithm::run()
   pvoid lStarKtcAlgorithm run(LSTARKTCALGORITHM T* this){
      // InactivateLinksInTriangles
                                                                                     InactivateLinksInTriangles
      void** result2 black = pattern InactivateLinks black(this);
      while (result2 black != NULL) {
        LINK T^* e12 = (LINK T^*) result2 black[5];
        free (result2 black);
        void** result3 green = pattern InactivateLinks green(e12);
        free (result3 green);
        // Possible nested scopes
10
        result2 black = pattern InactivateLinks black(this);
      } // End of InactivateLinksInTriangles
11
12
                                                                                            [Failure]
                                        Traceability via story node names
13
      // ActivateRemainingLinks
                                                                                      ActivateRemainingLinks
      void** result4 black = pattern ActivateRemainingLinks black(this);
      while (result4 black != NULL) {
        LINK T^* e12 = (LINK T^*) result4 black[1];
        free (result4 black);
        void** result5 green = pattern ActivateRemainingLirks green(e12);
        free (result5 green);
        result4 black = pattern ActivateRemainingLinks black(this);
      } // End of ActivateRemainingLinks
                                       Story pattern decomposed into sub-
 Generic match representation:
                                                                                            [Failure]
```

patterns, e.g., for the LHS and RHS

Example: Pattern matching code (I)



```
for (e13 = list head pred(list e13 this node outgoingLinks, this node, & node isOutgoingLinks);
     e13!=NULL;
     e13=list item next pred(e13,this node, &node isOutgoingLinks)) {
 NODE T* n3 = link getTarget(e13);
                                               Proper NULL
  if (n3 != NULL) {
                                                                                                      n3: Node
                                               pointer handling
    if (!node equals(n3, this node)) {
                                                                                                                           out,
      LINK T* e12;
                                                                                  e13: Link
                                                                                                                        e32: Link
      list t list e12 this node outgoingLinks = node getOutgoingLinks (this
                                                                                                   LStarKTCAlgorithm
      for (e12 = list head pred(list e12 this node outgoingLinks, this node
                                                                                                                          √trg
          e12!=NULL;
                                                                                                      e12: Link
          e12=list item next pred(e12,this node,&node isOutgoingLinks)) {
                                                                                  n1: Node
                                                                                                  state == UNCL.
                                                                                                                        n2: Node
        if (!link equals(e12, e13)) {
                                                                                                  state := INACTIVE
                                                          Mimic object
          orientation by
                                                                                  max(maxWeight, e13.weight, e32.weight)
          if (n2 != NULL) {
                                                                                  min(minWeight, e13.weight, e32.weight)
                                                           conventional
            if (!node equals(n2, this node)) {
                                                                                  *(kMinWeight, minWeight, this.k)
                                                           names
              if (!node equals(n2, n3)) {
                                                                                  >(e12.weight, maxWeight); >(e12.weight, kMinWeight)
                LinkState e12 marked = link getMarked(e12);
                                                                                  hopCountOK(e12.hopCount, e13.hopCount, e32.hopCount, a)
                if(linkState equals(e12 marked, UNCLASSIFIED)){
                   EInt n2 hopcount = node getHopcount(n2);
                   EInt n3 hopcount = node getHopcount(n3);
                  if(lStarKtcAlgorithm evaluateHopcountConstraint(
                     this node hopcount , n2 hopcount , n3 hopcount , this stretchFactor )) {
User-defined
                   LINK T* e32;
I*-kTC
                  list t list e32 n3 outgoingLinks = node getOutgoingLinks(n3);
constraint
                  for (e32 = list head pred(list e32 n3 outgoingLinks,n3,&node isOutgoingLinks);
                        e32!=NULL:
                        e32=list_item_next_pred(e32,n3,&node isOutgoingLinks)) {
                     if (!link equals(e13, e32)) {
```

if (!link equals(e12, e32)) {

Example: Pattern matching code (II)

}}}}}



```
for (e32 = list head pred(list e32 n3 outgoingLinks,n3,&node isOutgoingLinks);
     e32!=NULL:
     e32=list item next pred(e32,n3,&node isOutgoingLinks)) {
  if (!link equals(e13, e32)) {
    if (!link equals(e12, e32)) {
      if (node containsIncomingLinks(n2, e32)) {
        EDouble e12 weight = link getWeight(e12);
        EDouble e13 weight = link getWeight(e13);
        EDouble e32 weight = link getWeight(e32);
        EDouble maxWeight = e13 weight < e32 weight ? e32 weight : e13 weight ;
        if(e12 weight > maxWeight){
          EDouble minWeight = e13 weight < e32 weight ? e13 weight : e32 weight ;
          EDouble kMinWeight = minWeight *this k ;
          if(e12 weight > kMinWeight){
            void** result = (void**) malloc(7*sizeof(
                                                                                     n3: Node
             if( result == NULL) {
                                                                                                          .out
               printf("ERROR[topologycontrol]: could no
                                                                e13: Link
                                                                                                      e32: Link
                                                                                 LStarKTCAlgorithm
               return NULL;
                                                                                                         Λin
             }else{
                                                                                                         √trg
               result[0]= this;
                                                                                    e12: Link
                                                                n1: Node
                                                                                state == UNCL.
                                                                                                      n2: Node
               result[1] = this node;
                                                                                state := INACTIVE
               result[2]= n2;
               result[3]= n3;
                                                                max(maxWeight, e13.weight, e32.weight)
                                                                min(minWeight, e13.weight, e32.weight)
               result[4]= e13;
                                   Memory mgmt. (II)
                                                                *(kMinWeight, minWeight, this.k)
               result[5]= e12;
                                                                >(e12.weight, maxWeight); >(e12.weight, kMinWeight)
                                  malloc on match,
               result[6]= e32;
                                                                hopCountOK(e12.hopCount, e13.hopCount, e32.hopCount, a)
                                  notification on error
               return result;
```

Evaluation for code size



... instead of runtime/scalability because (code) memory is scarcer (ca. 48kB)!

Algorithm	Image Size[B]	Δ rel. to NoTC	Δ rel. to Man.
No TC	36 917-	"Boilerplate:"	OS + sample application
kTC Man	39 135	+2 218 (+ 6.0 %)	+1 762 (+ 79.2%)
Gen	40 897	+3 980 (+10.8%)	
I*-kTC Man	40 293	+3 376 (+ 9.1%)	+1 954 (+ 57.9%)
Gen	42 247	+5 330 (+14.4%)	
LMST Man	39 395	+2 478 (+ 6.7%)	+3 404 (+137.4%)
Gen	42 799	+5 882 (+15.9%)	

Increase relative to image size: +4.8pp..+9.2pp



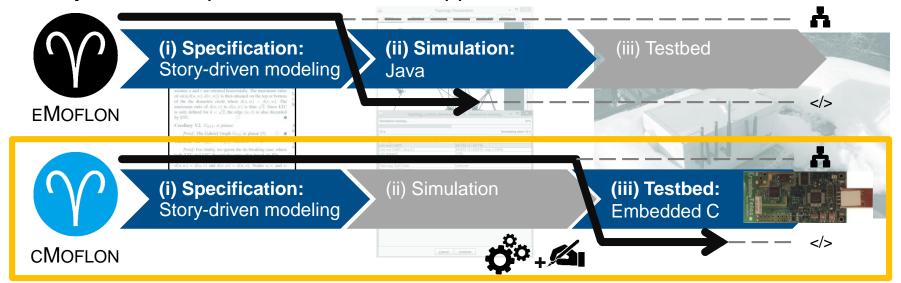
Summary and Outlook



Generate embedded C code of Topology Control algorithms



- Extensible, dynamic memory & null-pointer handling
- Compatible with previous simulation support



Outlook

- More sensor platforms and algorithms
- Improve memory allocation behavior
- Reduce redundant null-pointer checks





Thank You for Your Attention



Department of Electrical Engineering and Information Technology Real-Time Systems Lab



Roland Kluge, M.Sc.

roland.kluge@es.tu-darmstadt.de Merckstraße 25 64282 Darmstadt

Germany

Phone: +49(0)6151 16-22354 Fax: +49(0)6151 16-22352 www.es.tu-darmstadt.de



Icon sources



- eMoflon/cMoflon logos: Work of the eMoflon developer team. Subject to Fair Use conditions.
- TelosB Sensor node: Wiki Commons, CC-BY-3.0, by Jbasic, https://commons.wikimedia.org/wiki/File:TelosB.jpg
- PowWow Sensor node: Wiki Commons, CC-BY-3.0, by Romain Fontaine, https://commons.wikimedia.org/wiki/File:PowWow.jpg
- Writing Hand: CC-BY-3.0 US, Rediffusion, 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
- Barrier: CC-BY-3.0 US, ProSymbols US, https://thenounproject.com/search/?q=barrier&i=792199
- Lock: CC-BY-3.0 US, Aleksandr Vector, https://thenounproject.com/search/?q=lock&i=1009759
- Light Bulb: CC-BY-3.0 US, ProSymbols, https://thenounproject.com/search/?q=idea%20bulb%20icon&i=799621
- Target: CC-BY-3.0 US, Gregor Cresnar, https://thenounproject.com/search/?q=target&i=796213
- **GitHub Logo**: CC-BY-3.0, Dave Gandy, http://www.flaticon.com/free-icon/github-logo 25231

These slides are provided under Creative Commons (BY-SA,

