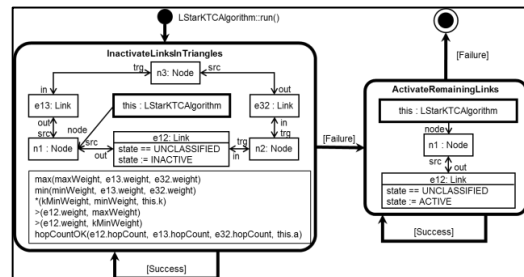
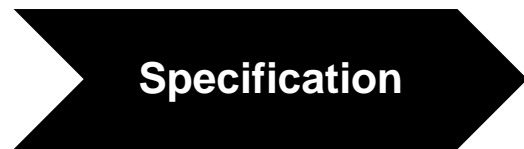


cMoflon: Model-Driven Generation of Embedded C Code for Wireless Sensor Networks

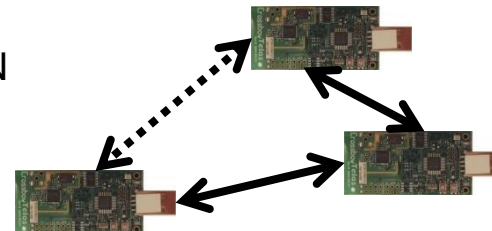
Conference Talk at ECMFA 2017
2017-07-19



TECHNISCHE
UNIVERSITÄT
DARMSTADT



cMOFLON



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

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

Proof: We will show that $(u, v) \notin G_{GG}$ implies that $(u, v) \notin G_{KTC}$. Pick a $(u, v) \in G - G_{GG}$. Then there must exist a $w \in G$ such that w lies inside the diametric circle of u and v . By the assumption of the UDG, $(u, v) \in G$ implies that $(u, w) \in G$ and $(v, w) \in G$. 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 the diametric circle where $d(u, w) = d(v, w)$. The maximum ratio of $d(u, v)$ to $d(u, w)$ is thus $\sqrt{2}$. Since KTC is only defined for $k < \sqrt{2}$, the edge (u, v) is also discarded by KTC. \square

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

Proof: We will show that $(u, v) \notin G_{GG}$ implies that $(u, v) \notin G_{KTC}$. Pick a $(u, v) \in G - G_{GG}$. Then there must exist a $w \in G$ such that w lies inside the diametric circle of u and v . By the assumption of the UDG, $(u, v) \in G$ implies that $(u, w) \in G$ and $(v, w) \in G$. 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 the diametric circle where $d(u, w) = d(v, w)$. The maximum ratio of $d(u, v)$ to $d(u, w)$ is thus $\sqrt{2}$. Since KTC is only defined for $k < \sqrt{2}$, the edge (u, v) is also discarded by KTC. \square

Corollary V.2. G_{KTC} is planar.

Proof: The Gabriel Graph G_{GG} is planar [7]. \square

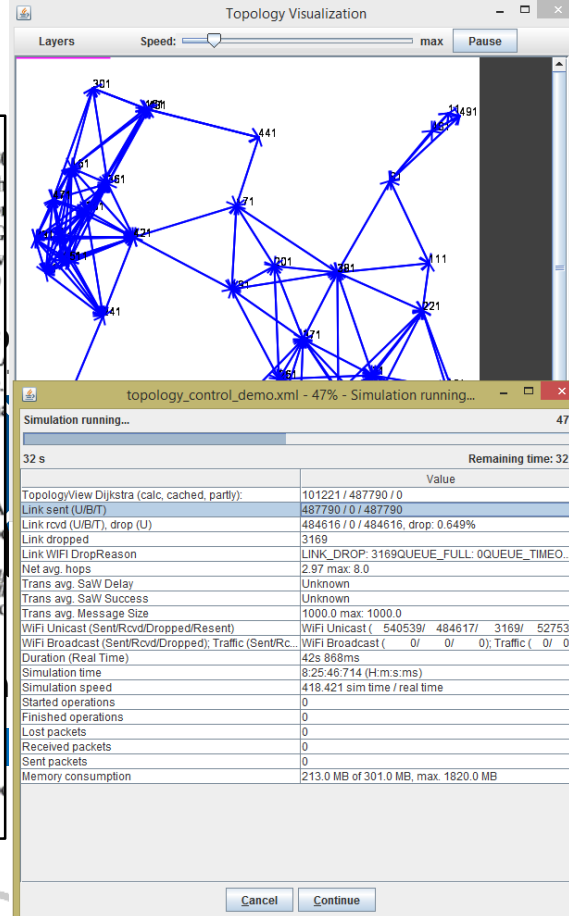
Theorem V.3. $G_{KTC} = G_{XTC}$ whenever $k = 1$.

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 iff 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 removes exactly the longest edge. \square

Corollary V.4. $G_{XTC} \subseteq G_{KTC}$

Proof: Increasing k only adds edges. \square



DFG Collaborative Research Centre 1053 – MAKI

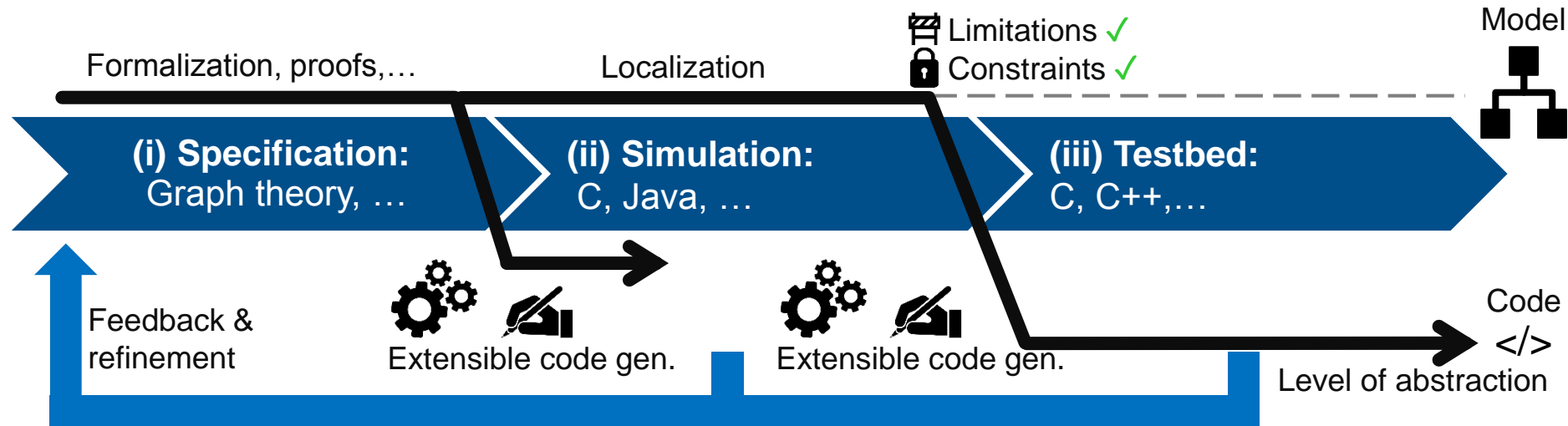
https://www.flocklab.ethz.ch/wiki/chrome/site/wiki_public/observer/outdoor_1.jpg



Leverage model-driven engineering principles!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

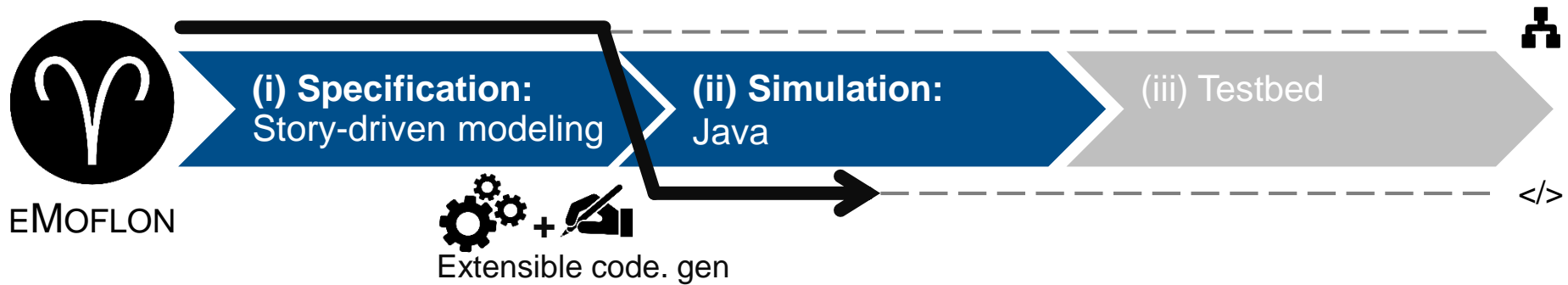


Contribution

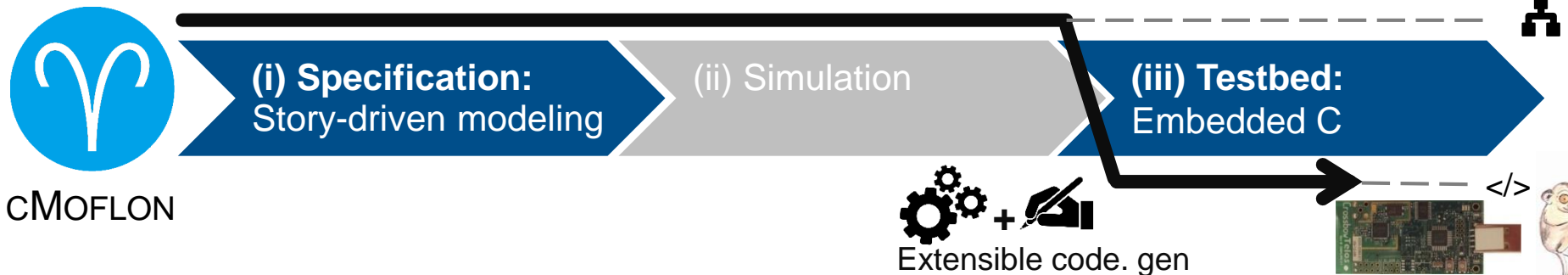


TECHNISCHE
UNIVERSITÄT
DARMSTADT

Previous: Integration with wireless network simulator



This paper: Code generation for wireless sensor testbed

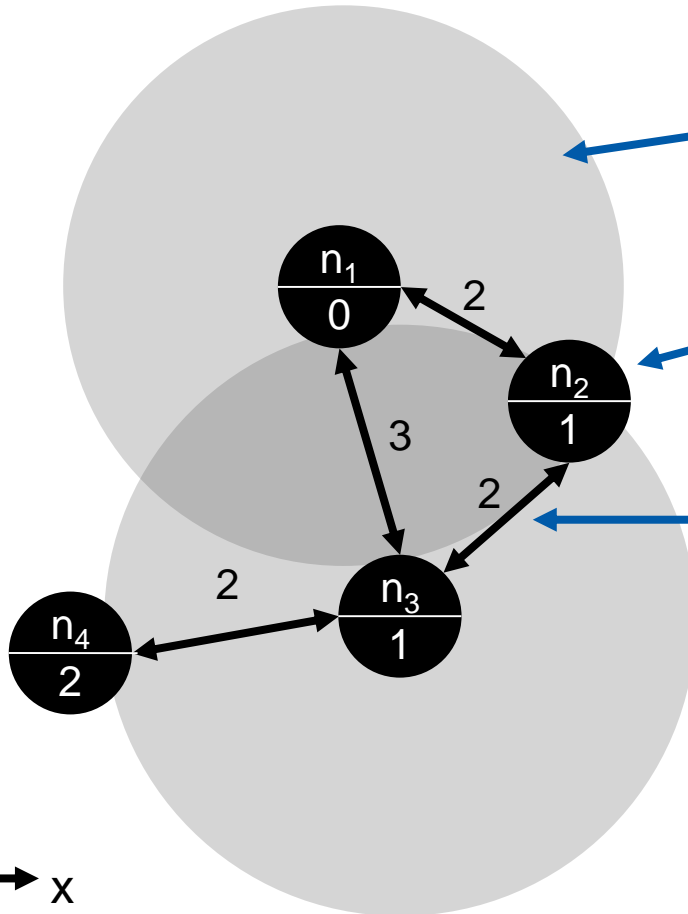


Wireless Sensor Networks



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wireless Sensor Network



Transmission range

Wireless node n_2
+ hop count $h_1(n_2)$ to n_1

Wireless link e_{34}
+ link weight $w(e_{34})$
(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



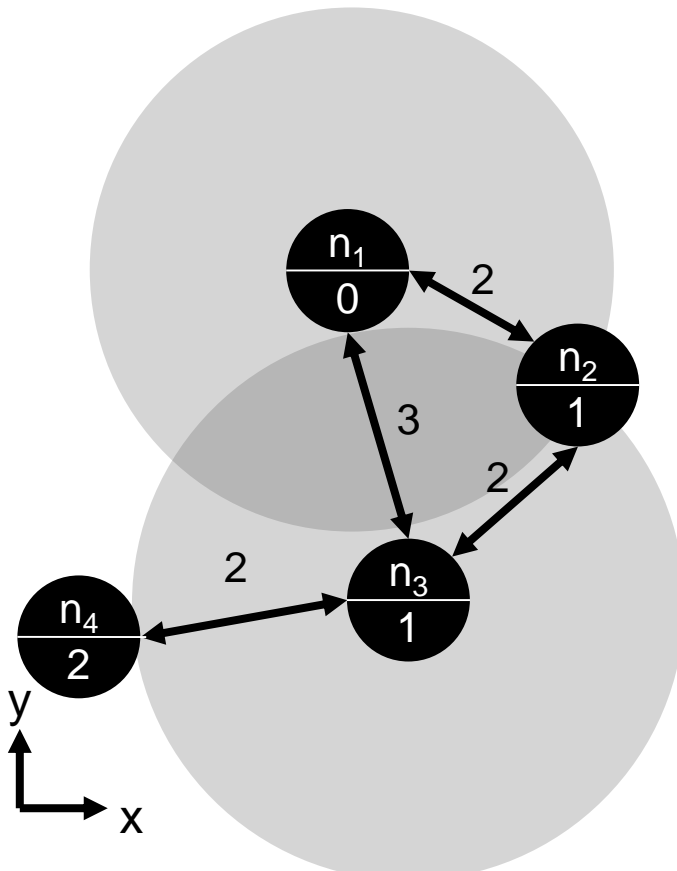
Topology Control

High weight means
high energy consumption

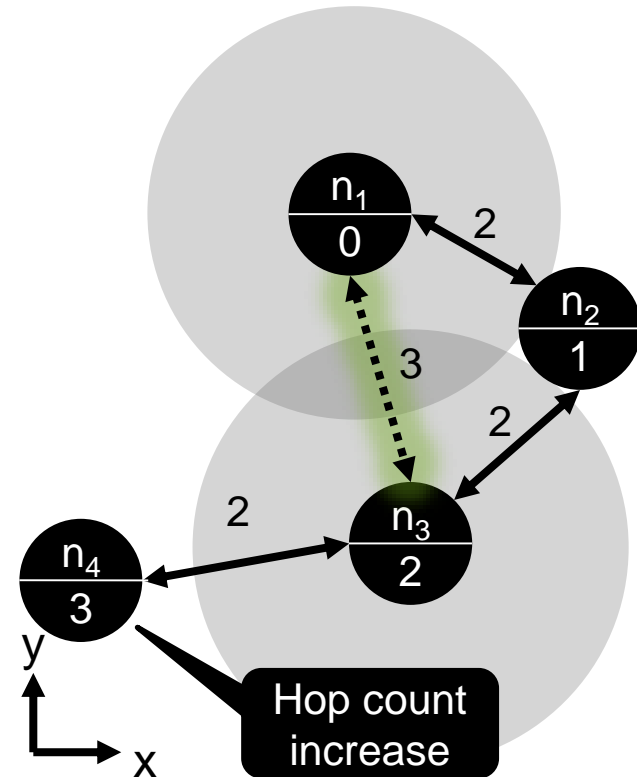


Inactivate high-
weight links

↔ Active
⋯ Inactive

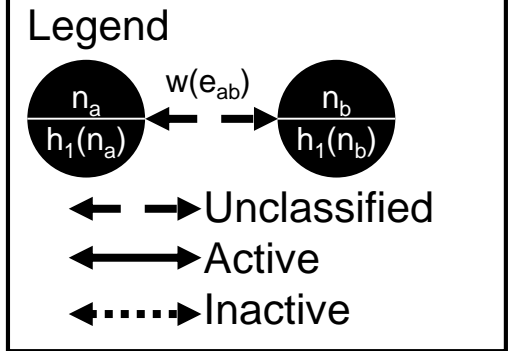
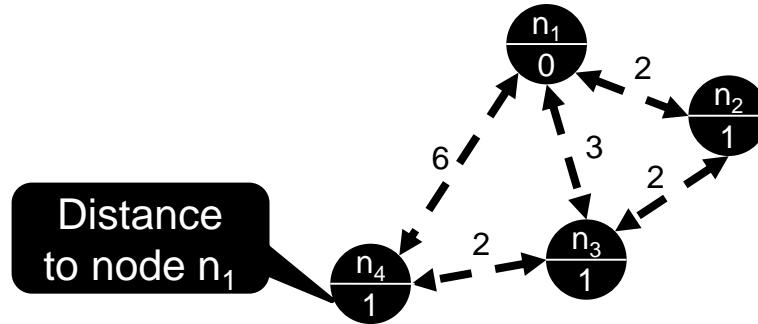


Topology Control

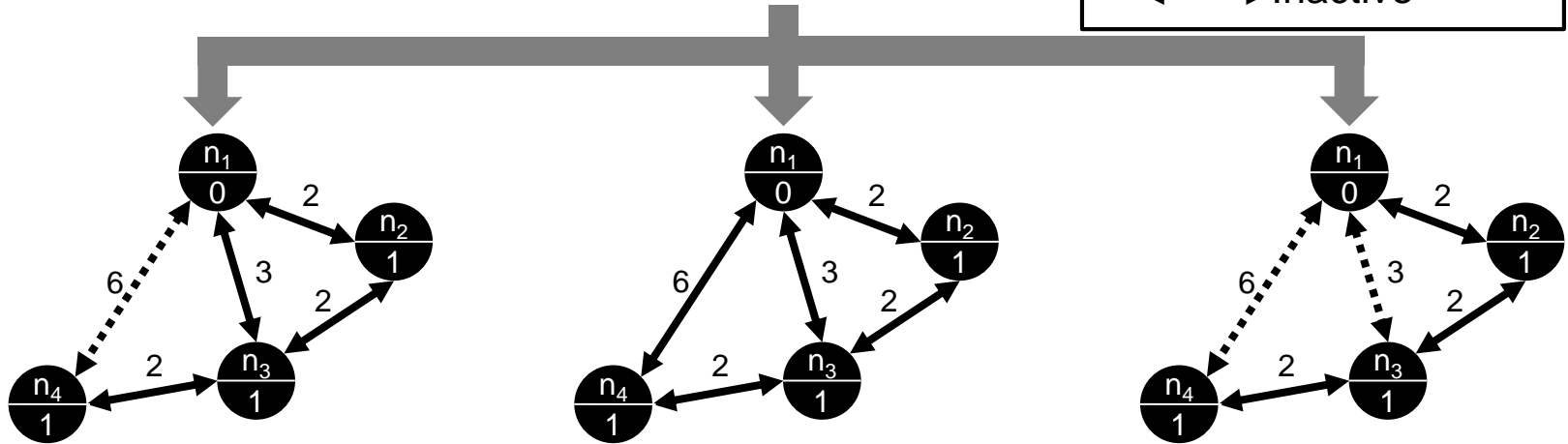


Running examples: kTC, I*-kTC, and LMST

Input topology



Output topology






kTC_[Schweizer16] (**k=2**)
Inactivate the weight-maximal 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 incurs 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

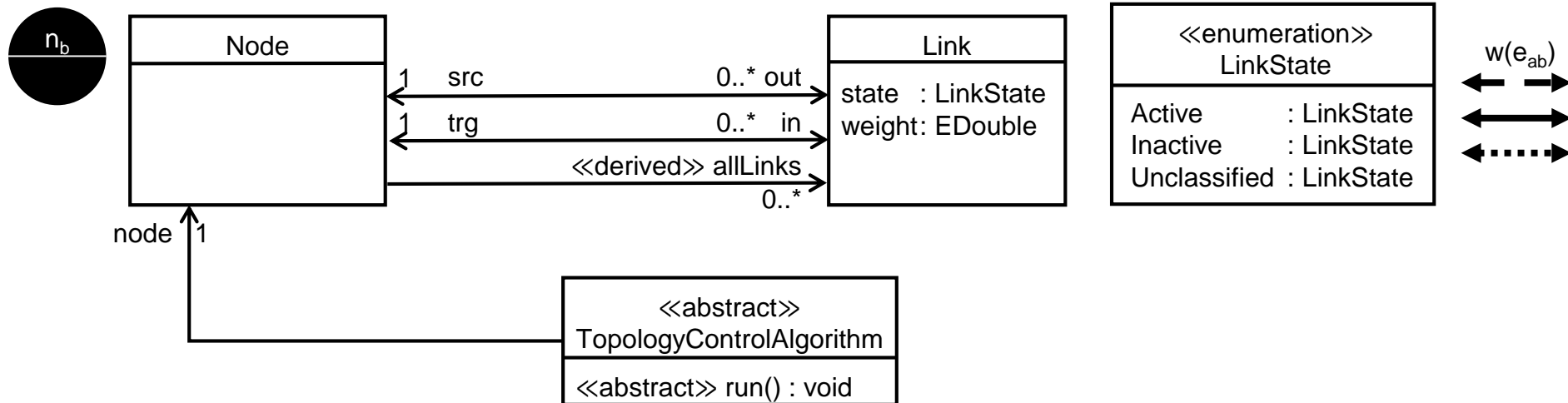


Topology Control metamodel

Spec.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

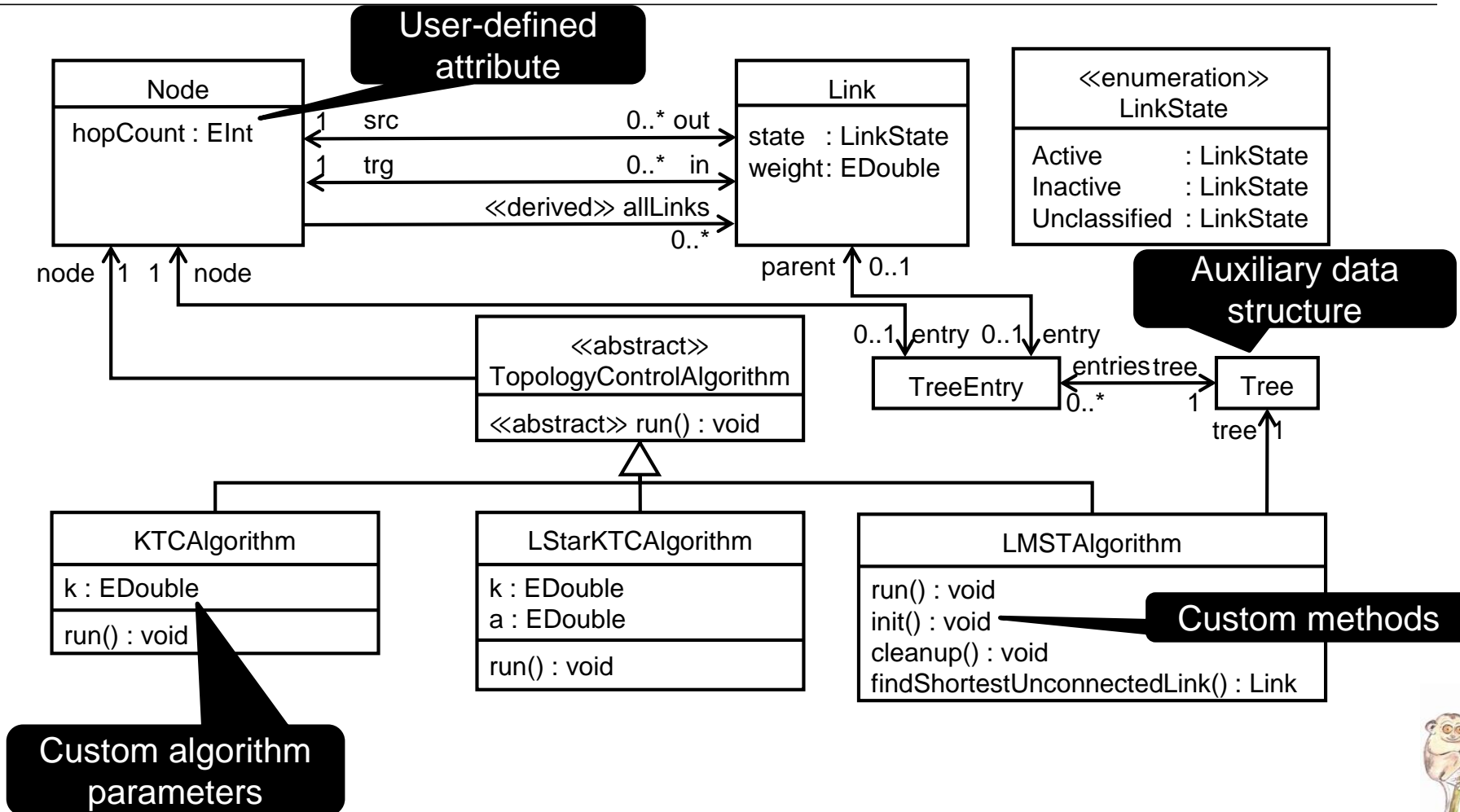


Topology Control Metamodel

Spec.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

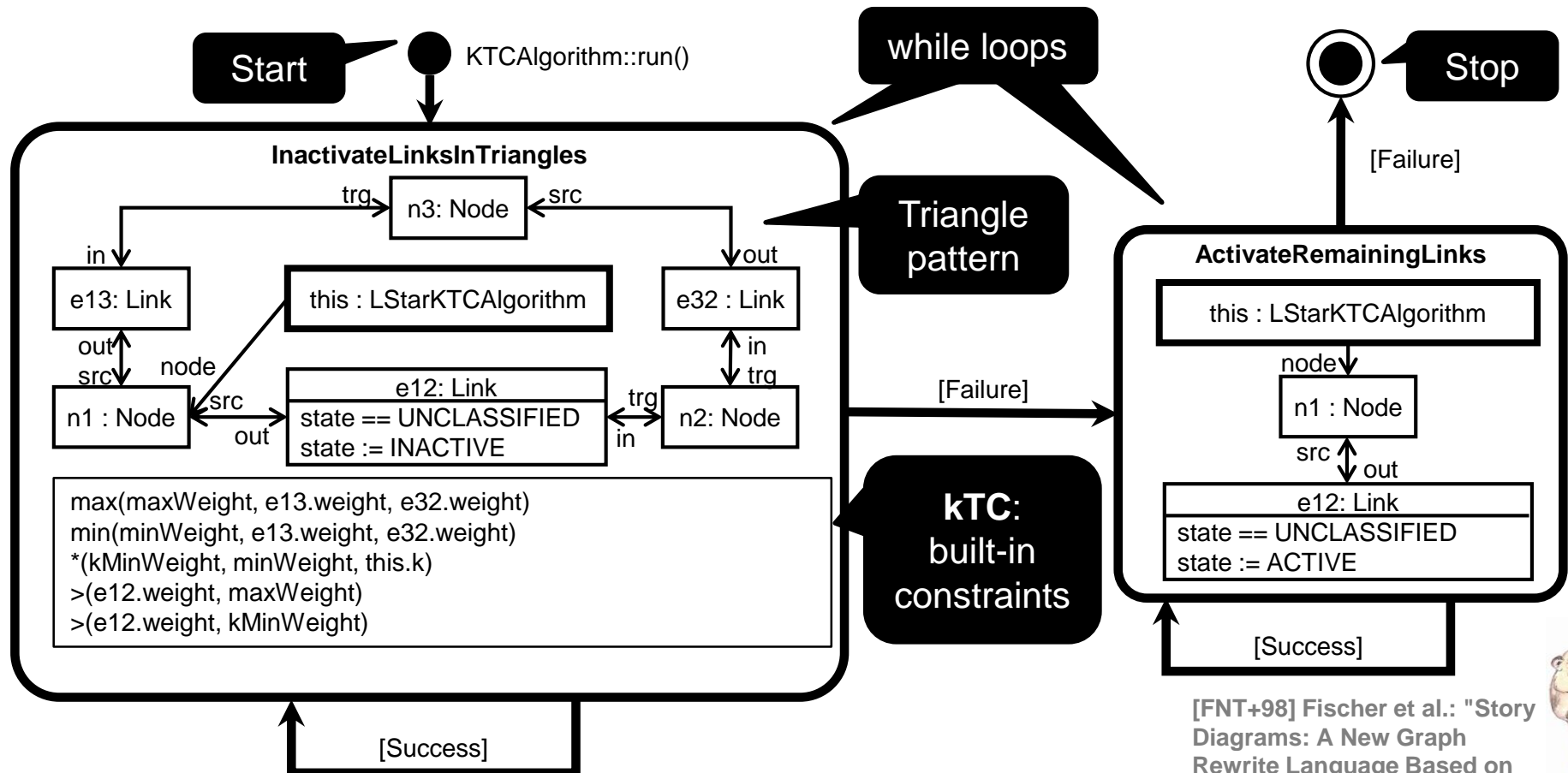


kTC story diagram

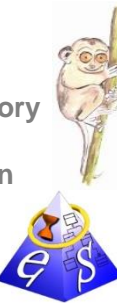
Spec.



TECHNISCHE
UNIVERSITÄT
DARMSTADT



[FNT+98] Fischer et al.: "Story Diagrams: A New Graph Rewrite Language Based on the UML and Java," 1998



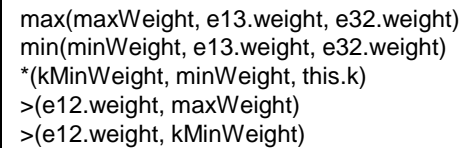
► Spec.



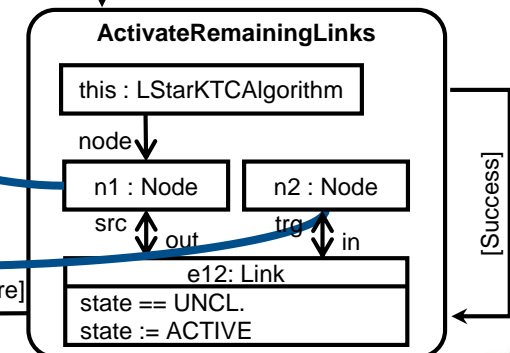
Initial



After loop
"InactivateLinksInTriangle"



After loop
"ActivateLinksInTriangle"



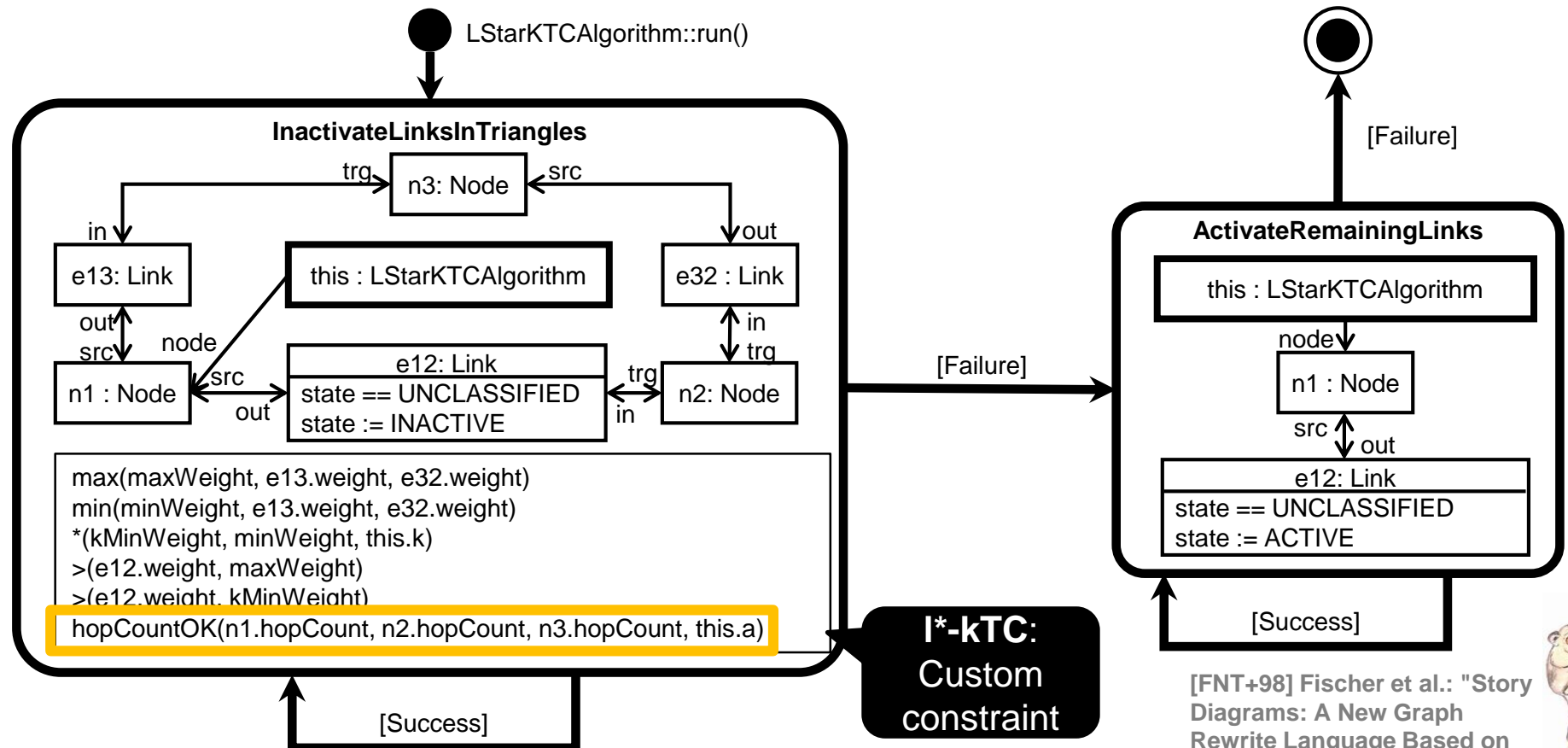
match (only for type Node)

I*-kTC story diagram

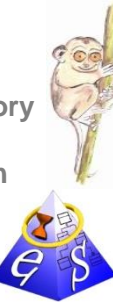
Spec.



TECHNISCHE
UNIVERSITÄT
DARMSTADT



[FNT+98] Fischer et al.: "Story Diagrams: A New Graph Rewrite Language Based on the UML and Java," 1998





CODE GENERATION FOR TESTBED EVALUATION

(i) Specification

(ii) Simulation

(iii) Testbed



CONTIKI and ToCoCo as target platform

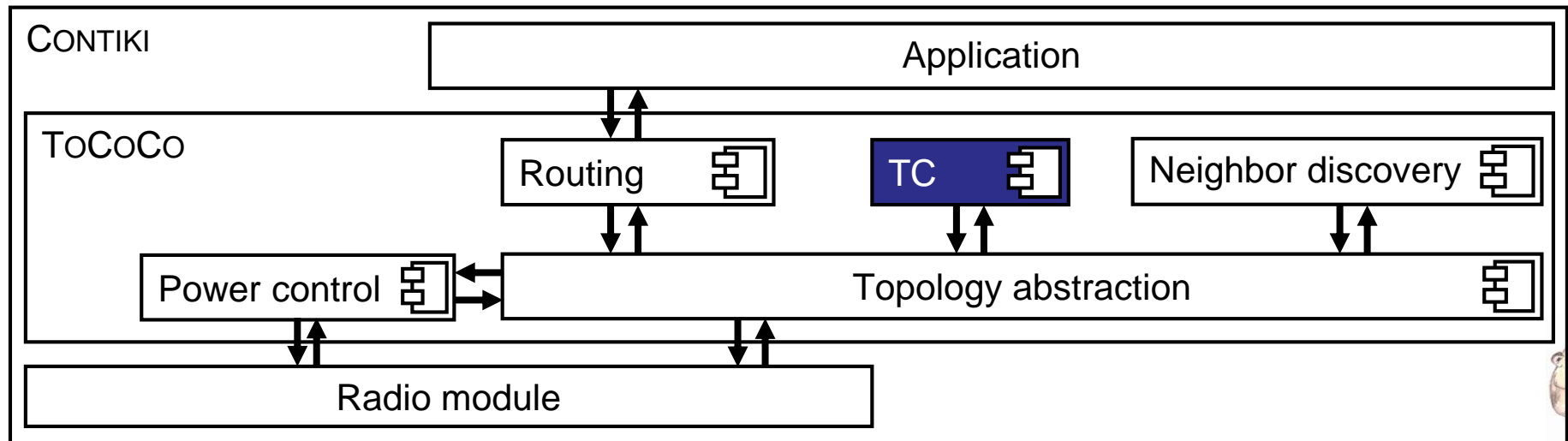


TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **Contiki OS**: widely used in the WSN community
- **ToCoCo – Topology Control framework for Contiki**: rapid prototyping and evaluation of Topology Control algorithms

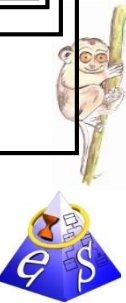
– **cMoflon creates a**  **per subclass of**

```
<<abstract>>  
TopologyControlAlgorithm  
<<abstract>> run() : void
```



[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>

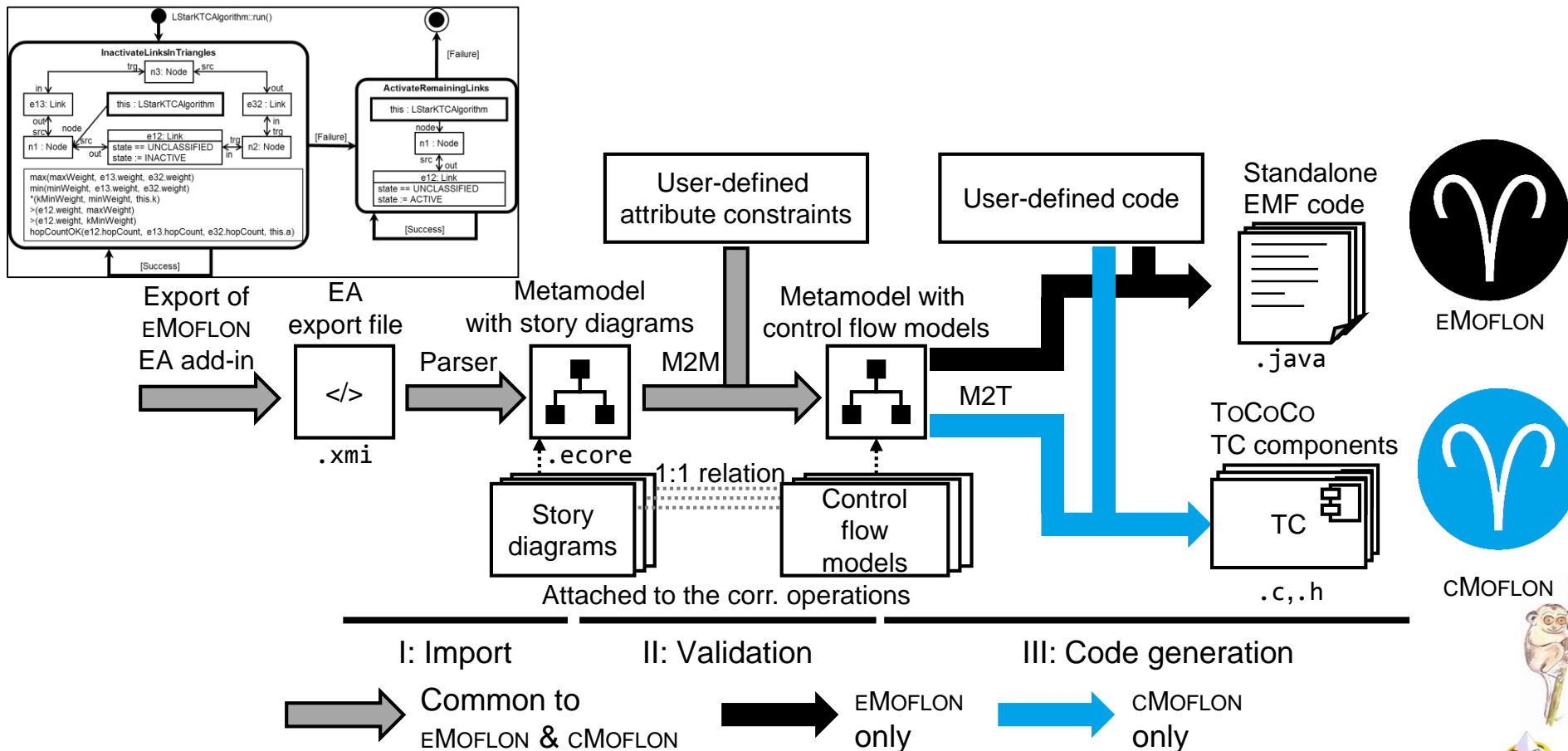


A lean variant of EMOFLON



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



Example: Control flow code

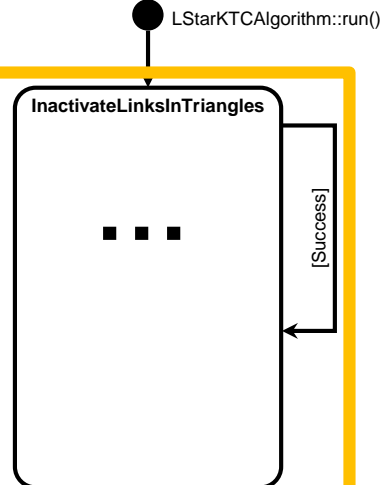


TECHNISCHE
UNIVERSITÄT
DARMSTADT

Memory management (I):
free after use

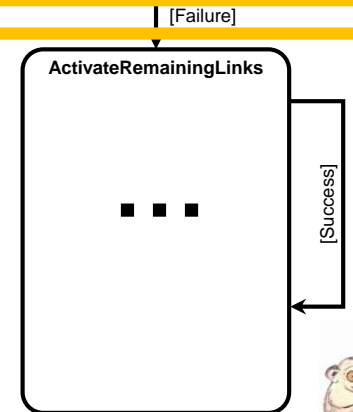
Auxiliary type: Developer provides
mapping (i.e., C struct)

```
1 void lStarKtcAlgorithm run(LSTARCTCALGORITHM T* this){
2 // InactivateLinksInTriangles
3 void** result2_black = pattern_InactivateLinks_black(this);
4 while (result2_black != NULL) {
5     LINK T* e12 = (LINK T*) result2_black[5];
6     free(result2_black);
7     void** result3_green = pattern_InactivateLinks_green(e12);
8     free(result3_green);
9     // Possible nested scopes
10    result2_black = pattern_InactivateLinks_black(this);
11 } // End of InactivateLinksInTriangles
```



```
13 // ActivateRemainingLinks
14 void** result4_black = pattern_ActivateRemainingLinks_black(this);
15 while (result4_black != NULL) {
16     LINK_T* e12 = (LINK_T*) result4_black[1];
17     free(result4_black);
18     void** result5_green = pattern_ActivateRemainingLinks_green(e12);
19     free(result5_green);
20     result4_black = pattern_ActivateRemainingLinks_black(this);
21 } // End of ActivateRemainingLinks
```

Traceability via story node names



Generic match representation:
Array of void*

Story pattern decomposed into sub-
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);
    if (n3 != NULL) {
```

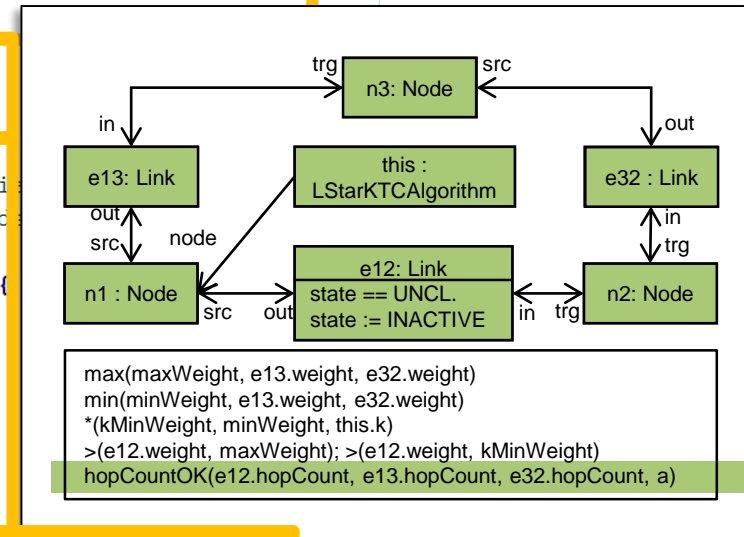
Proper NULL
pointer handling

```
        if (!node_equals(n3, this_node)) {
            LINK_T* e12;
            list_t list_e12_this_node_outgoingLinks = node_getOutgoingLinks(this_node);
            for (e12 = list_head_pred(list_e12_this_node_outgoingLinks, this_node);
                e12 != NULL;
                e12 = list_item_next_pred(e12, this_node, &node_isOutgoingLinks)) {
                if (!link_equals(e12, e13)) {
                    NODE_T* n2 = link_getTarget(e12);
                    if (n2 != NULL) {
```

Mimic object
orientation by
conventional
names

User-defined
l*-kTC
constraint

```
                    if (!node_equals(n2, this_node)) {
                        if (!node_equals(n2, n3)) {
                            LinkState e12_marked = link_getMarked(e12);
                            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)) {
                                    LINK_T* e32;
                                    list_t list_e32_n3_outgoingLinks = node_getOutgoingLinks(n3);
                                    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)

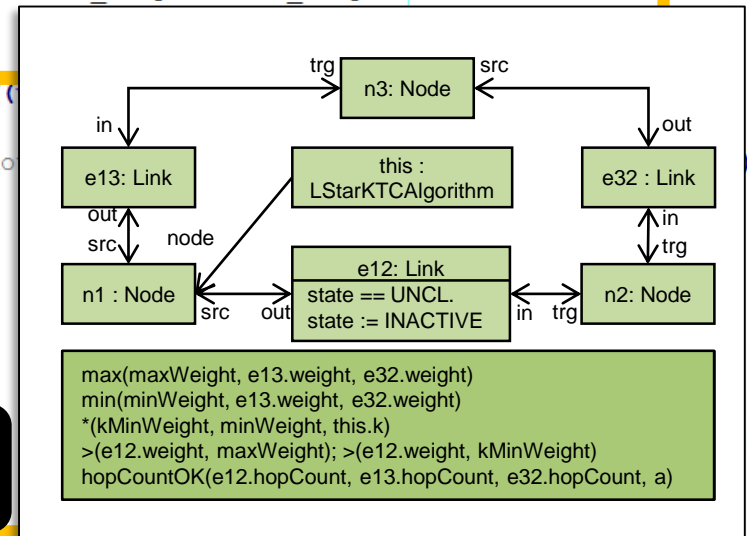


```

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(sizeof(
                        if(_result == NULL){
                            printf("ERROR[topologycontrol]: could no
                            return NULL;
                        }else{
                            _result[0]= _this;
                            _result[1]= this_node;
                            _result[2]= n2;
                            _result[3]= n3;
                            _result[4]= e13;
                            _result[5]= e12;
                            _result[6]= e32;
                            return _result;
                    }
            }
        }
    }
}

```

Memory mgmt. (II)
malloc on match,
notification on error



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

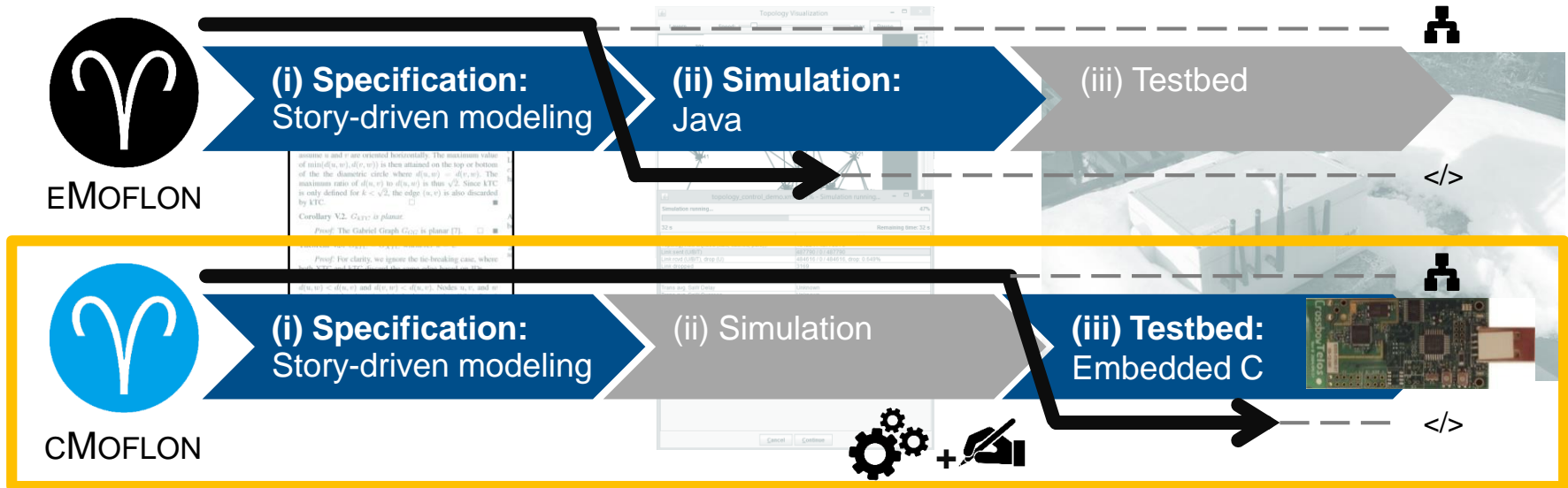


TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Generate **embedded C code** of Topology Control algorithms
- **Extensible**, dynamic memory & null-pointer handling
- **Compatible** with previous simulation support



Open source:
[eMoflon/cmoflon](https://github.com/EMOFLON/cmoflon)



Outlook

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

Thank you!



Thank You for Your Attention



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **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,

