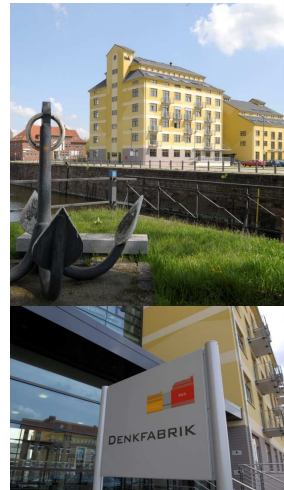
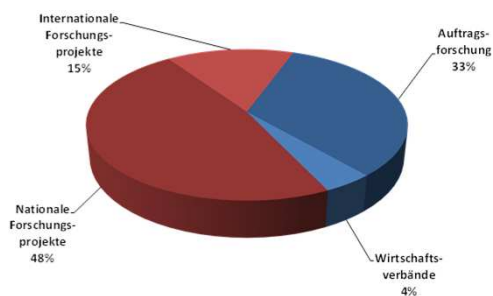


Vorstellung ifak

- ifak: Institut für Automation und Kommunikation e.V., Magdeburg
- 50 Mitarbeiter plus Studenten und Gastwissenschaftler
- Zusammensetzung der Erträge aus F&E:



3

ifak

Geschäftsfelder des ifak

IKT & Automation

- Echtzeitkommunikation, Geräte- und Systemtest...



Wasser & Energie

- Kläranlagen, Integrierte Planungswerkzeuge...



Messtechnik & Leistungselektronik

- Prozessmesstechnik, Kontaktlose Energie- und Datenübertragung...



Verkehr & Assistenz

- Intelligente Verkehrssysteme, Qualitätssicherung vernetzter Mobilitätssysteme, Elektromobilität...

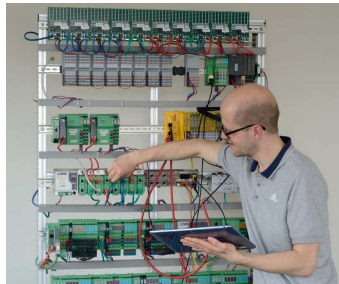


4

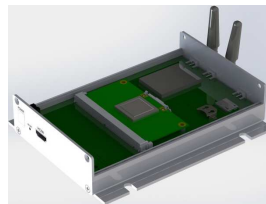
ifak

Einige Kommunikations-Domänen

- Kommunikation in der Automation
 - Ethernet, PROFINET...
 - Prüflabor Feldbussysteme



- Kommunikation im Verkehr
 - WLANp, Car2X...
 - Kooperative Applikationen



5

ifak

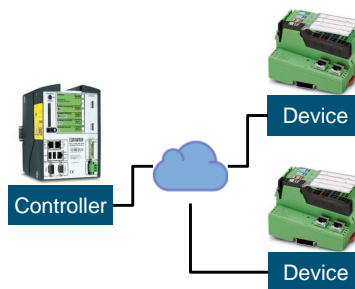
Motivation: Testen von Netzwerkkommunikation

6

ifak

Motivation: Industrieautomation mit **PROFINET**-Protokoll

- Steuerungen (Controller) tauschen Daten mit Feldgeräten (Devices) aus
- Typische Testzeitpunkte:
 - Bei Inbetriebnahme einer Anlage
 - Nach Aufspielen neuer Stack-Versionen
- Typische Testziele:
 - Erfolgreicher Verbindungsaufbau, ggf. zu mehreren Devices
 - Robustheit gegenüber Störungen



7

ifak

Motivation

- Wunsch: Manuelle Testabläufe **automatisieren**
- Bsp.: PROFINET-Verbindungsaufbau in Wireshark:

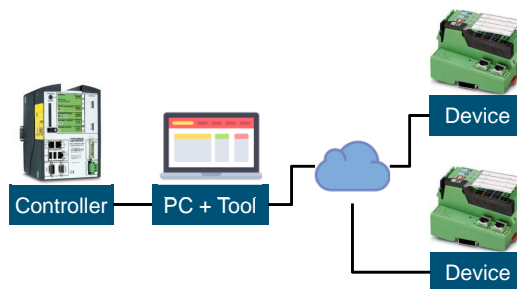
Time	Source	Destination	Protocol	Length	Info
12.087517	PhoenixC_511e:be	PN-MC_00:00:00	PN-DCP	60	Ident Req, Xid:0x120d13ad, NameOfStation:"bk01"
12.613632	PhoenixC_511e:be	PN-MC_00:00:00	PN-DCP	60	Ident Req, Xid:0x120e13ae, AliasName:"port-001.rfc-470-pn-51-1e-be"
14.190131	PhoenixC_511e:be	PN-MC_00:00:00	PN-DCP	60	Ident Req, Xid:0x120f13af, NameOfStation:"bk01"
14.196615	PhoenixC_19:2c:4f	PhoenixC_511e:be	PN-DCP	122	Ident ok, Xid:0x120f13af, NameOfStation:"bk01", Dev-Options(9), DeviceVendorValue
14.219544	PhoenixC_19:2c:50	LLDP_Multicast	LLDP	119	Chassis Id = bk01 Port Id = port-001 TTL = 20 RTClass3 Port Status = OFF
14.224106	PhoenixC_511e:c0	LLDP_Multicast	LLDP	114	Chassis Id = rfc-470-pn-51-1e-be Port Id = port-001 TTL = 20 RTClass3 Port Status
15.303088	PhoenixC_511e:be	Broadcast	ARP	60	Who has 172.16.48.200? Tell 172.16.48.206
15.306568	PhoenixC_19:2c:4f	PhoenixC_511e:be	ARP	60	172.16.48.200 is at 00:a0:45:19:2c:4f
16.374337	PhoenixC_511e:c0	LLDP_Multicast	LLDP	114	Chassis Id = rfc-470-pn-51-1e-be Port Id = port-001 TTL = 20 RTClass3 Port Status
17.446753	PhoenixC_511e:be	Broadcast	ARP	60	Who has 172.16.48.200? Tell 172.16.48.206
17.450708	PhoenixC_19:2c:4f	PhoenixC_511e:be	ARP	60	172.16.48.200 is at 00:a0:45:19:2c:4f
17.453146	172.16.48.206	172.16.48.200	PNIO-CM	641	Connect request, ARBLockReq, IOCRBLockReq, IOCRBLockReq, ExpectedSubmoduleLockReq
17.475674	172.16.48.200	172.16.48.206	PNIO-CM	244	Connect response, OK, ARBLockRes, IOCRBLockRes, IOCRBLockRes, AlarmCRBLockRes, Mo
17.570402	172.16.48.206	172.16.48.200	PNIO-CM	212	Write request, IOWriteReqHeader, Api:0x0, Slot:0x0/0x1, Index:(0x1), 6 bytes
17.578695	172.16.48.200	172.16.48.206	PNIO-CM	206	Write response, OK, IOWriteResHeader, Api:0x0, Slot:0x0/0x1, Index:(0x1), OK
17.599673	PhoenixC_511e:be	PhoenixC_19:2c:4f	PNIO	60	RTCI(legacy), ID:0xc010, Len: 40, cycle:65504 (valid,Primary,ok,Run)
17.632142	172.16.48.206	172.16.48.200	PNIO-CM	209	Write request, IOWriteReqHeader, Api:0x0, Slot:0x0/0x1, Index:(0x2), 3 bytes
17.638700	172.16.48.200	172.16.48.206	PNIO-CM	206	Write response, OK, IOWriteResHeader, Api:0x0, Slot:0x0/0x1, Index:(0x2), OK
17.682703	PhoenixC_19:2c:4f	PhoenixC_511e:be	PNIO	60	RTCI(legacy), ID:0xc000, Len: 40, cycle:57312 (valid,Primary,ok,Run)
17.694034	172.16.48.206	172.16.48.200	PNIO-CM	215	Write request, IOWriteReqHeader, Api:0x0, Slot:0x2/0x1, Index:(0x7000), 9 bytes
17.709654	172.16.48.200	172.16.48.206	PNIO-CM	206	Write response, OK, IOWriteResHeader, Api:0x0, Slot:0x2/0x1, Index:(0x7000), OK
17.753729	172.16.48.206	172.16.48.200	PNIO-CM	215	Write request, IOWriteReqHeader, Api:0x0, Slot:0x3/0x1, Index:(0x7000), 9 bytes
17.762605	172.16.48.200	172.16.48.206	PNIO-CM	206	Write response, OK, IOWriteResHeader, Api:0x0, Slot:0x3/0x1, Index:(0x7000), OK
17.848499	172.16.48.206	172.16.48.200	PNIO-CM	174	Control request, IOControlReq Pfm End.req, Command: ParameterEnd
17.855028	PhoenixC_511e:be	PhoenixC_19:2c:4f	PNIO	60	RTCI(legacy), ID:0xc010, Len: 40, cycle:8160 (valid,Primary,ok,Run)
17.860667	172.16.48.200	172.16.48.206	PNIO-CM	174	Control response, OK, IOControlRes Pfm End.rsp, Command: done
17.883689	172.16.48.200	172.16.48.206	PNIO-CM	206	Control request, IOXBLockReq Application Ready.req, Command: ApplicationReady, Mo
17.931900	172.16.48.206	172.16.48.200	PNIO-CM	174	Control response, OK, IOXBLockRes Application Ready.rsp, Command: done
17.938683	PhoenixC_19:2c:4f	PhoenixC_511e:be	PNIO	60	RTCI(legacy), ID:0xc000, Len: 40, cycle:65504 (valid,Primary,problem,run)
18.041997	172.16.48.206	172.16.48.200	PNIO-CM	206	Read request, IOReadReqHeader, Api:0x0, Slot:0x0/0x0, Index:PDRealData, 8192 byt
18.053681	172.16.48.200	172.16.48.206	PNIO-CM	418	Read response, OK, IOReadResHeader, Api:0x0, Slot:0x0/0x0, Index:PDRealData, 212
18.110248	PhoenixC_511e:be	PhoenixC_19:2c:4f	PNIO	60	RTCI(legacy), ID:0xc010, Len: 40, cycle:16332 (valid,Primary,ok,Run)
18.154683	PhoenixC_19:2c:4f	PhoenixC_511e:be	PNIO	60	RTCI(legacy), ID:0xc000, Len: 40, cycle:8160 (valid,Primary,problem,run)
18.365608	PhoenixC_511e:be	PhoenixC_19:2c:4f	PNIO	60	RTCI(legacy), ID:0xc010, Len: 40, cycle:24544 (valid,Primary,ok,Run)

8

ifak

Lösung: Testsoftware

- Projekt: PROGES (2012 – 2014) - „Programmierbarer Fehlergenerator für Ethernet-basierte Automatisierungsnetze“
- Anforderungen:
 - Erfüllung von Sollverhalten nachweisen
 - Gezielt Datenverkehr manipulieren
- Lösung: PC mit Testsoftware als Man-in-the-middle direkt hinter Steuerung
 - Überwachung
 - Manipulation



9

ifak

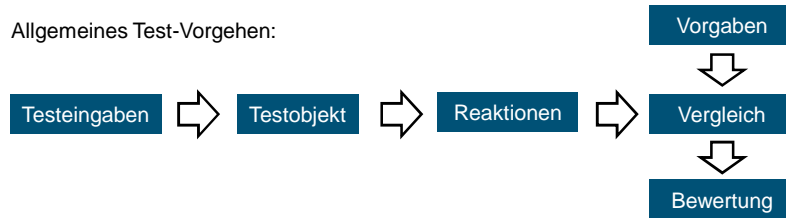
Ansatz: Modellbasierter Test
von sequentiellm Datenverkehr

10

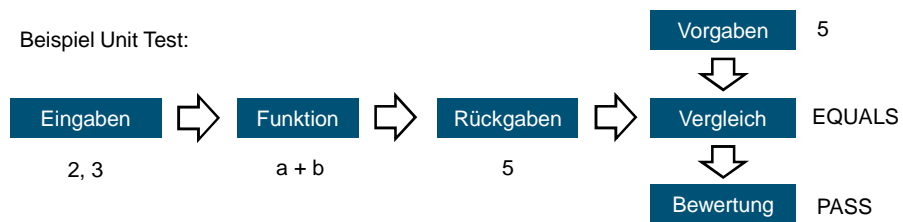
ifak

Ansatz: Testen der beobachtbaren Reaktionen

Allgemeines Test-Vorgehen:



Beispiel Unit Test:

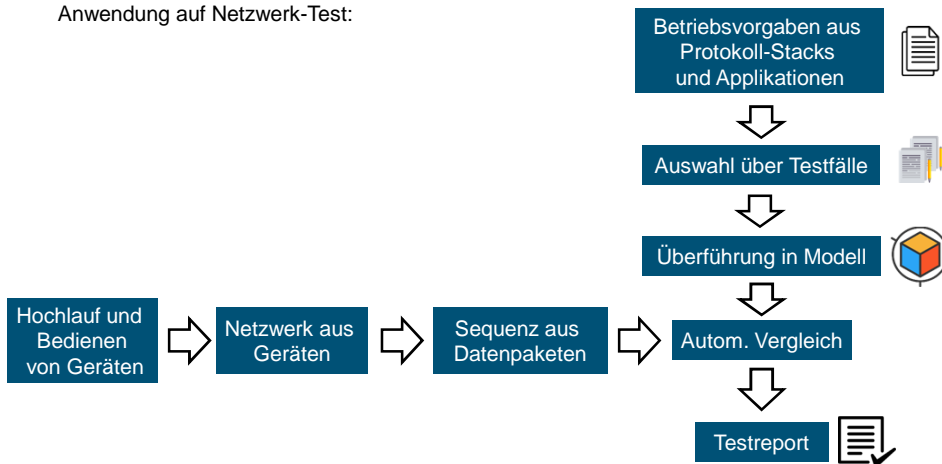


11

ifak

Ansatz: Testen der beobachtbaren Reaktionen

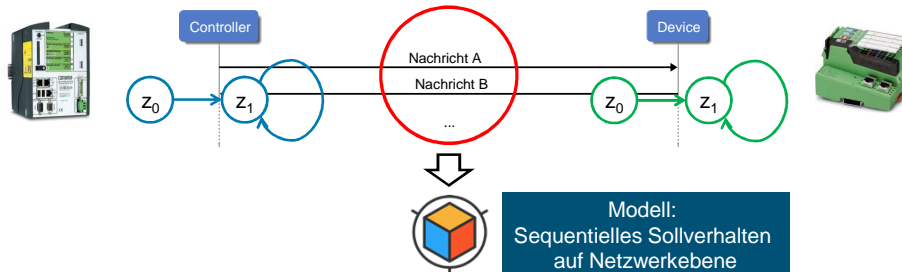
Anwendung auf Netzwerk-Test:



12

ifak

Ansatz: Testen der beobachtbaren Reaktionen



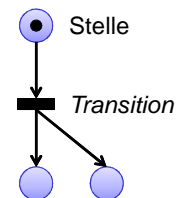
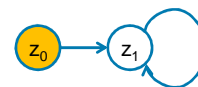
- Protokolle wie PROFINET über Automatenmodelle im Stack implementiert
- Sequentieller Datenverkehr beobachtbar
- Vollständiger Test aller Automaten aufgrund der Komplexität nicht möglich
- Alternative: Test ausgesuchten Datenverkehrs

13

ifak

Ansatz: Sollverhalten als Modelle

- Zustand in Sequenzdiagrammen nicht speicherbar
- Daher Überführung in gerichteten Graphen
- Netzwerk-Nachrichten ändern Zustand
- **Endlicher Zustandsautomat (FSM)**: Nur 1 Zustand zur gleichen Zeit. Jedoch Modellieren paralleler Komm.-Beziehungen gefordert.
- **Bedingungs-Ereignis-Netz** (einfachster Typ)
- **Stellen-Transitions-Netz** (Stellen mit erhöhter Kapazität)
- **Petrinetz** (Marken mit Attributen)

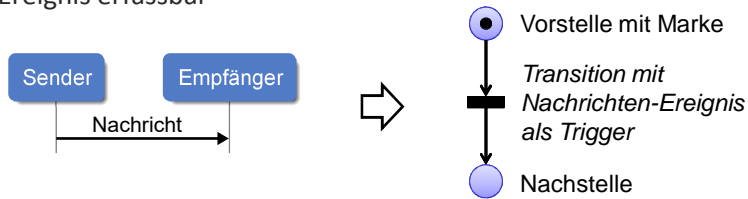


14

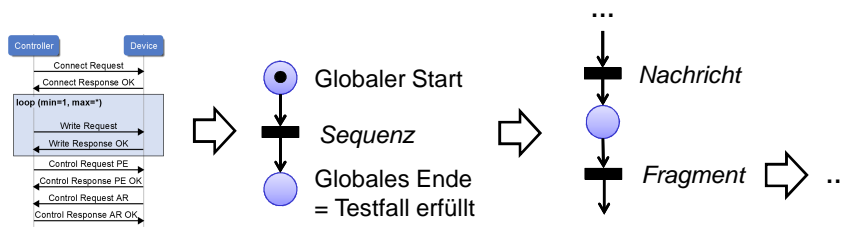
ifak

Ansatz: Sollverhalten als Modelle

- Sender und Empfänger nicht explizit modellieren, da nur Sende-Ereignis erfassbar



- Top-Down-Entwurf

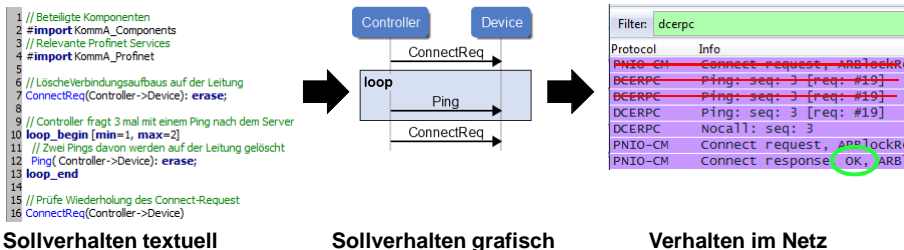


15

ifak

Manipulation des Datenverkehrs

- Weiteres Feature: Manipulation zur Fehlergenerierung
 - Fehlerhafte Geräte nachstellen
 - Protokoll-Stack prüfen (Reaktion bei Paketverlust etc.)
- Umsetzung: Action an Transition
 - Weiterleiten
 - Löschen (nicht weiterleiten)
 - Ändern von Parametern



Sollverhalten textuell

Sollverhalten grafisch

Verhalten im Netz

16

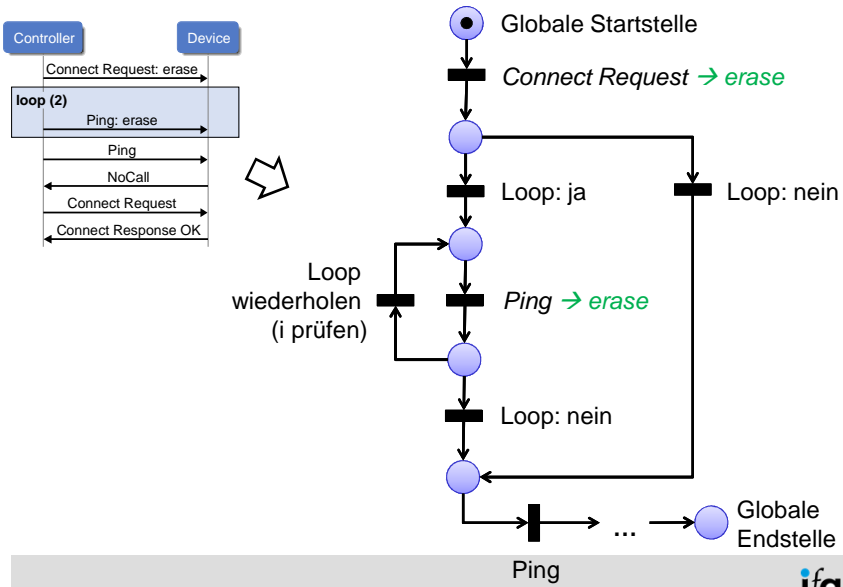
ifak

Markenspiel an einem Beispiel

17

ifak

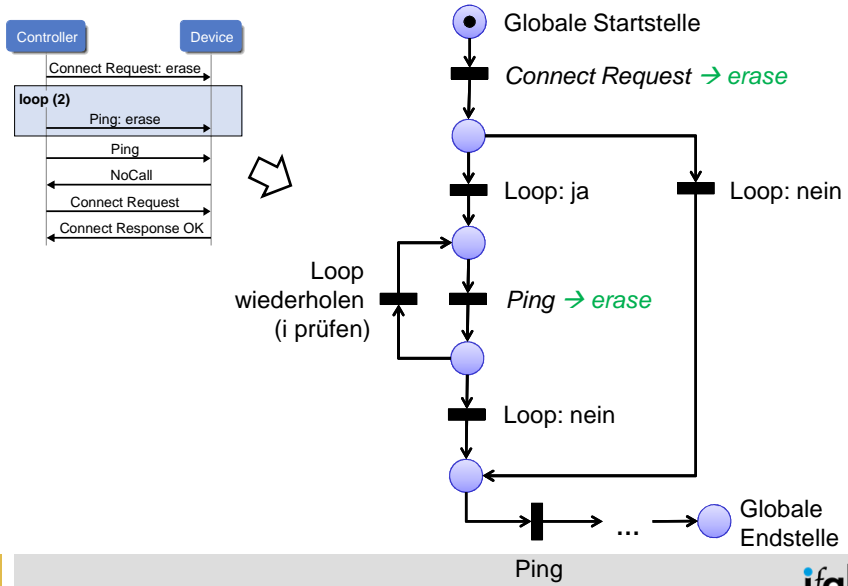
Beispiel Markenspiel



18

ifak

Beispiel Markenspiel



Fähigkeiten und Grenzen des Ansatzes

Fähigkeiten und Grenzen der Lösung

- Gutverhalten
 - Passiv mitschneiden
 - Punkt des Datenabgriffs frei wählbar (z. B. über TAP), hinter Steuerung am geeignetsten
- Negativverhalten
 - Gezielte Manipulation
 - Man-in-the-middle nötig
 - Zusätzliche Jitter und Latenzen
- Nur ausgewählte Testfälle: Automatisierung bisher manueller Prozesse – keine Abdeckung der gesamten Spezifikation
- Netzwerk-Tests ergänzend zu applikativen Tests

21

ifak



Automatisiertes Testen von verteilten Systemen über Petrinetze

Bei Fragen – einfach fragen!

ifak

Grafiken: <http://www.flaticon.com>

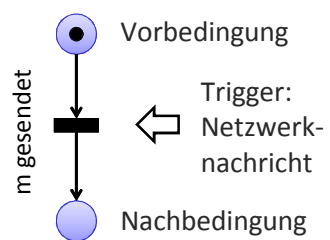
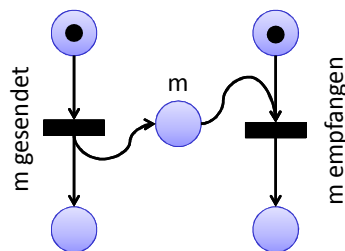
Zusatz-Folien

23

ifak

Modellierung von Kommunikationspartnern?

Sender: sendebereit Empfänger: empfangsbereit



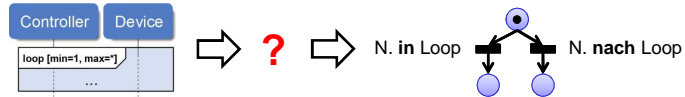
Problem: Beobachtbarkeit des Empfangsereignisses

24

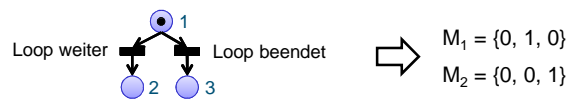
ifak

Post-Event-Entscheidbarkeit

- Bisher: **Entscheidbarkeit durch Netzwerkeignisse** (z. B. Loop beenden, Alternative entscheiden etc.)



- Durch zusätzliche Stellen & Transitionen nicht mehr (einfach) möglich
- Idee: **Mehrere, gleichzeitig aktive Markierungen**
- Dann erzeugen, wenn zwei Transitionen feuern könnten



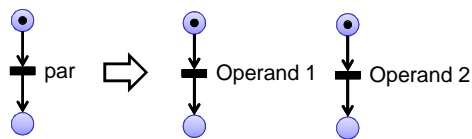
- Dann M_1 UND M_2 gegen Netzwerk-Nachrichten prüfen
- Eine M. „gewinnt“, sobald mit ihr weitere N.-Transition feuert

25

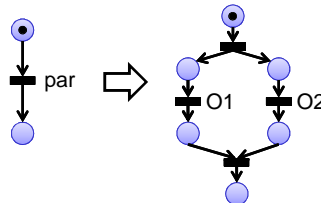
ifak

Überführung Sequenzen - Netze

- Anfangs:
 - Transitionen modellieren ausschließlich Netzwerk-Nachrichten
 - Dadurch minimale Zwischenstellen
 - Dadurch Fragmente in Fragmenten nie klar getrennt (eines beeinflusst ggf. das andere)



- Jetzt: Klare Abgrenzung: Zwischenstellen beginnen und beenden Fragmente



26

ifak