

Efficient Coordination with Semantic Shared Data Spaces

Präsentation der Dissertation

Martin Murth

Space Based Computing Group
Institut für Computersprachen, TU Wien



- Semantic Spaces
- Behandelte Problemstellungen
 - **RQ.1:** Koordination mit Semantic Spaces
 - **RQ.2:** Effiziente Event-Erkennung
 - **RQ.3:** Formale Beschreibung semantischer Event-basierter Systeme
- Evaluierung
- Zusammenfassung

Was ist ein Semantic Space?

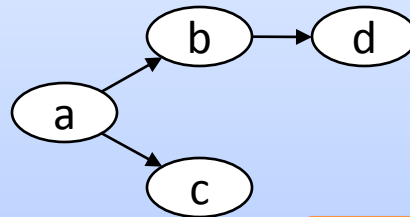
- Was ist ein „Semantic Space“?
 - ein gemeinsamer Datenraum?
 - ein Synchronisationsmechanismus?
 - eine Koordinationsplattform?
 - eine verteilte Middleware?
 - eine Wissensbasis?
 - eine Reasoning Engine?
- Ein Semantic Space ermöglicht
 - das **Speichern** von Information,
 - das **Ableiten** neuer Information anhand bestimmter Regeln und
 - das **Erkennen und Reagieren** auf neue/geänderte/gelöschte Information

- **Wo werden Semantic Spaces eingesetzt?**
 - Zur Realisierung wissensbasierter Anwendungen
 - Zum Lösen komplexer Koordinationsszenarien
 - Als Infrastrukturkomponente für Semantic Web

- **Wie funktioniert ein Semantic Space?**
 - *Ein Beispiel ...*

Client X

Semantic Space

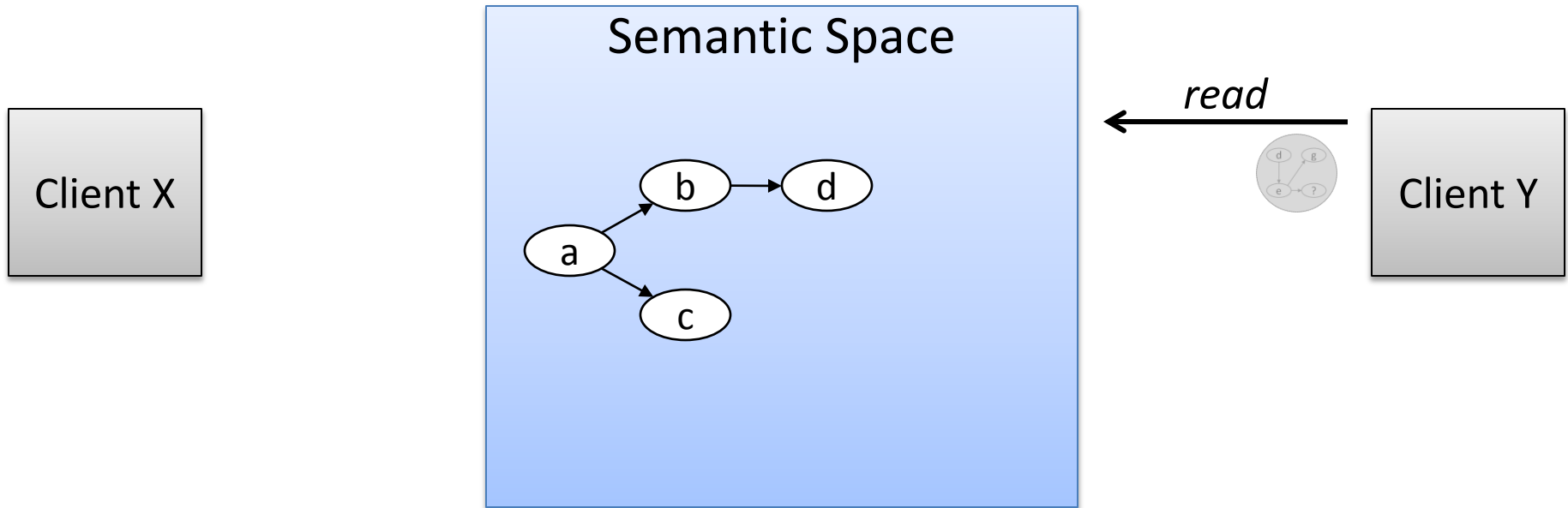


Client Y

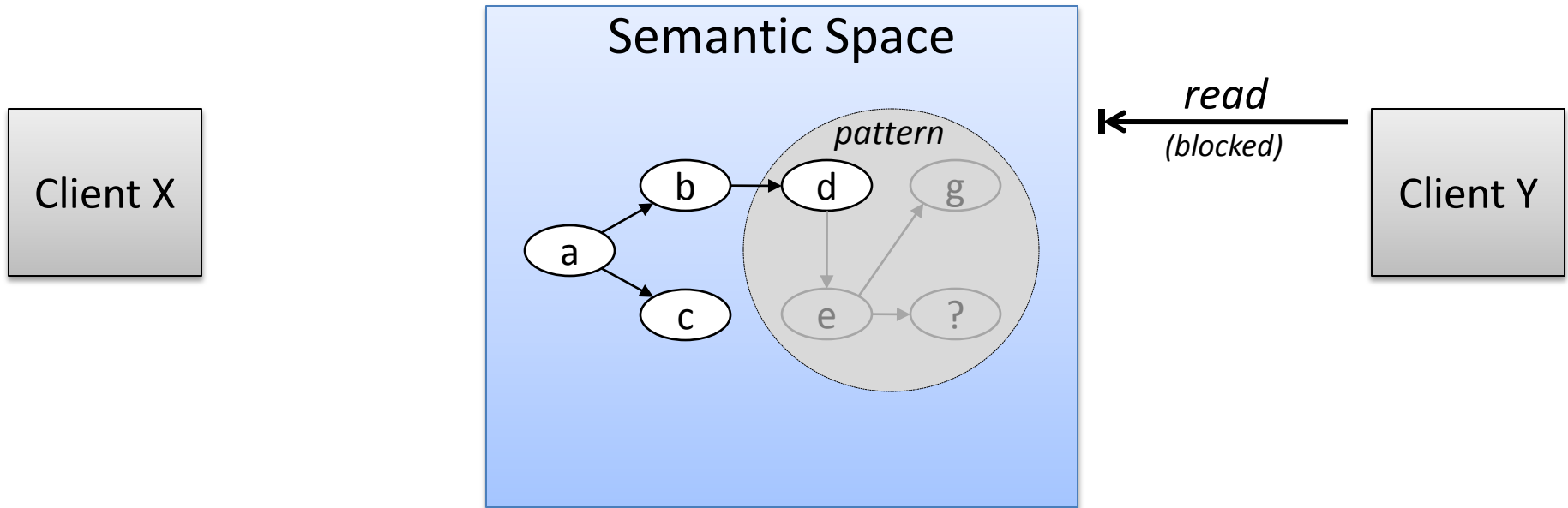
Interface

```
Graph read(String query);  
void publish(Graph graph);  
void delete(String query);  
SID subscribe(Subscriber s, String query);  
void unsubscribe(SID sid);
```

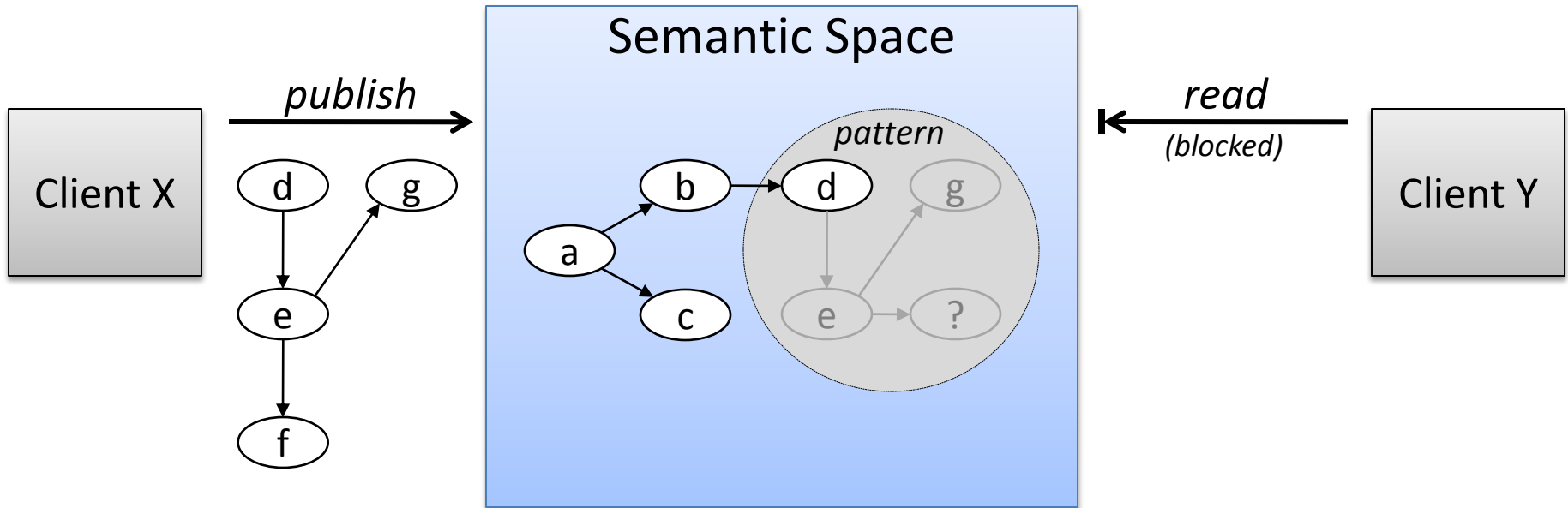
■ Read



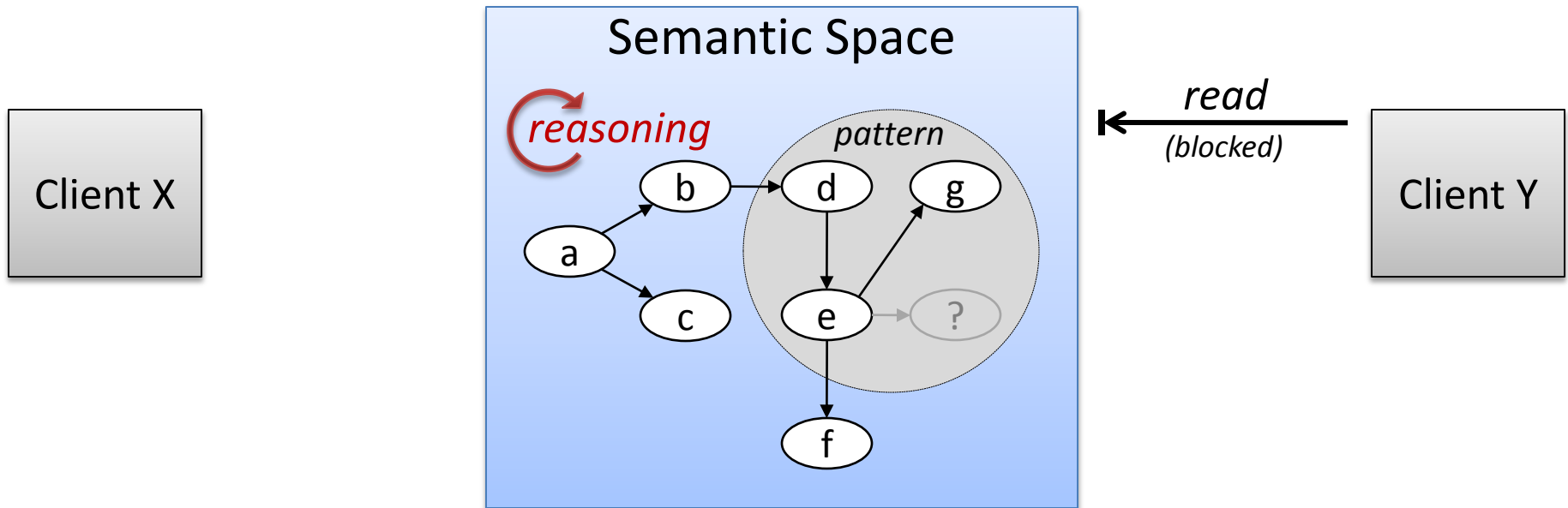
■ Read



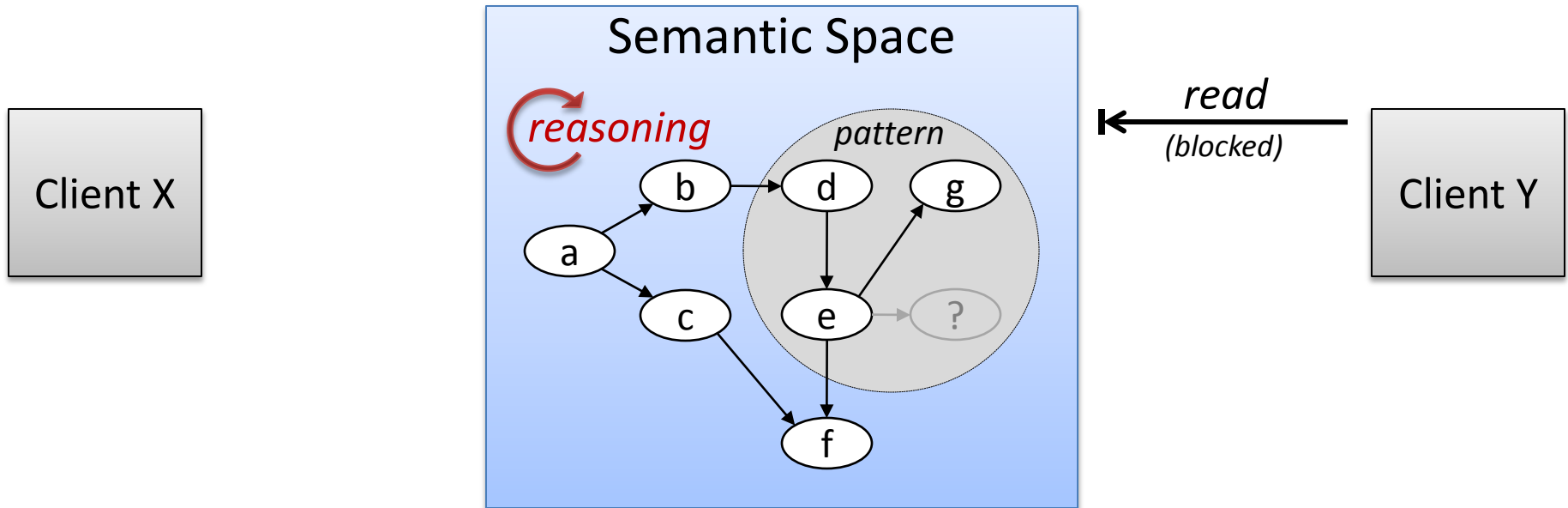
■ Publish



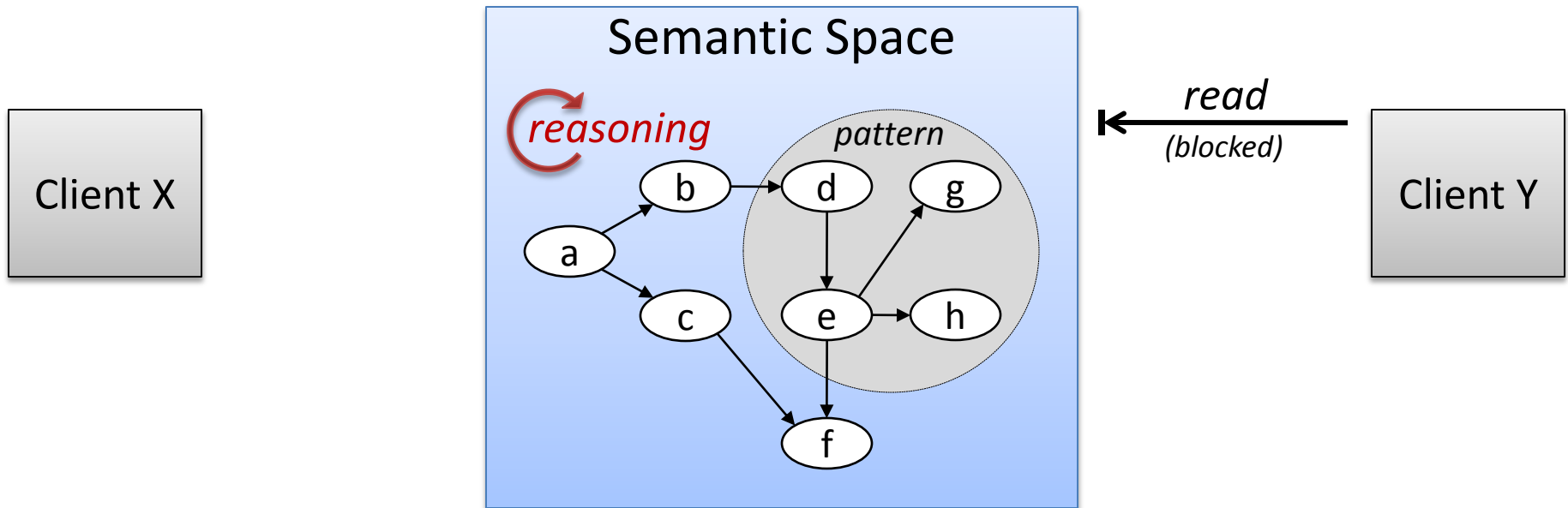
Reasoning



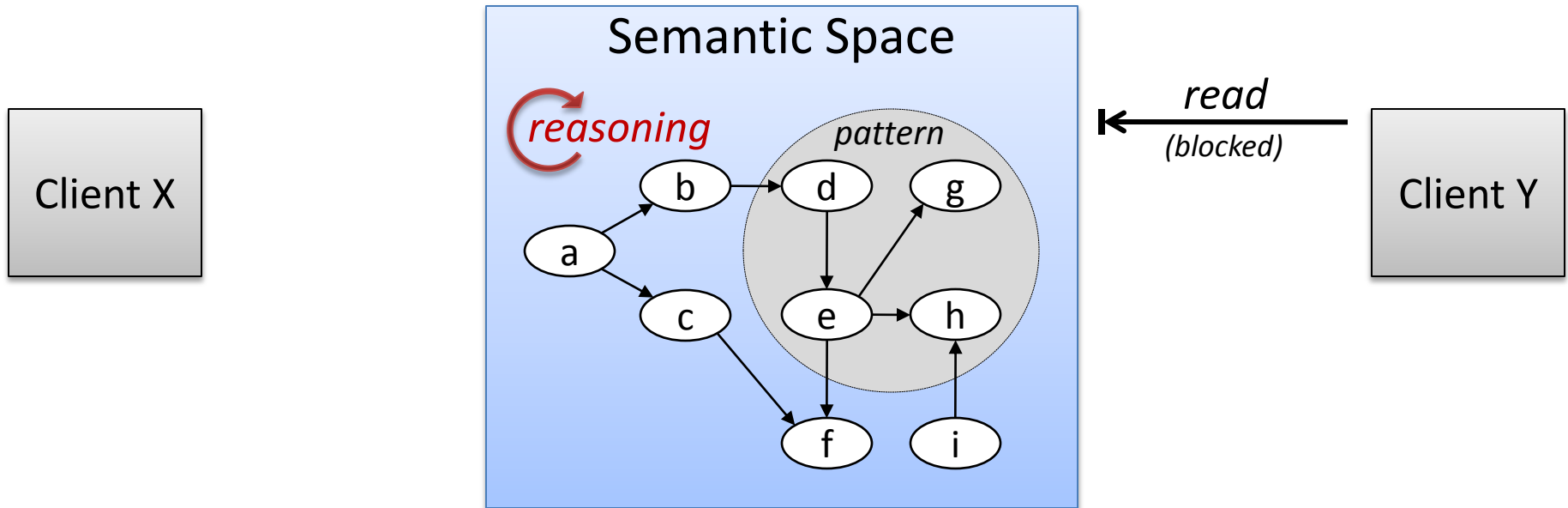
Reasoning



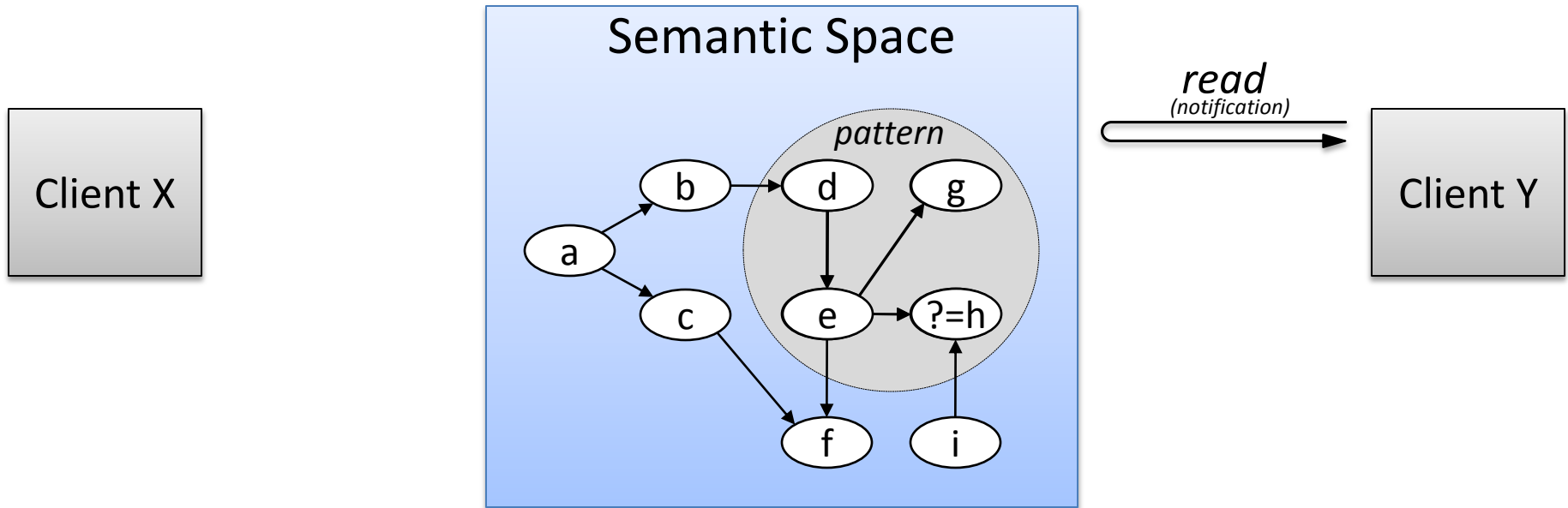
Reasoning



Reasoning



▪ Notify



Offene Probleme und Wissenschaftliche Fragestellungen

- Bisher wurden Semantic Spaces vorrangig im Kontext von Semantic Web Anwendungen untersucht
 - **RQ.1:** „Coordination with Semantic Spaces“
- Alle untersuchten Implementierungen weisen beträchtliche Performance-Einschränkungen auf
 - **RQ.2:** „Efficiency of Semantic Event Processing Systems“
- Es existieren keine formalen Methoden um das Verhalten von Semantic Spaces zu spezifizieren
 - **RQ.3:** „Formal Description of Semantic Spaces“

- Semantic Spaces
- Behandelte Problemstellungen
 - **RQ.1: Koordination mit Semantic Spaces**
 - RQ.2: Effiziente Event-Erkennung
 - RQ.3: Formale Beschreibung semantischer Event-basierter Systeme
- Evaluierung
- Zusammenfassung

RQ.1 Coordination with Semantic Spaces

■ Problemdefinition

- Semantic Spaces nur im Kontext von Semantic Web Anwendungen untersucht
- Oft nicht klar wie Koordination mit Semantic Spaces realisiert werden kann
- Oft auch nicht offensichtlich worin die Stärken und Schwächen dieses Ansatzes liegen

RQ.1 Coordination with Semantic Spaces.

What characterizes coordination with semantic spaces and what are the advantages and disadvantages of employing semantic technologies in space-based coordination systems?

- Vergleich und Analyse von
 - Semantic Spaces (TSC, TripCom, SWS, XVSM, sTuples)
 - Semantic Space Applikationen (e-Health, Enterprise Application Integration, etc.)
- Kriterien
 - Typische Verwendungsmuster
 - Konzeptuelle Vor- und Nachteile
 - Performance-Beobachtungen

RQ.1: Ergebnisse

- **Vorteile von Semantik in Space-basierten Koordinationsanwendungen**
 - + Hohe **Ausdruckskraft** der Koordinationssprache
 - + Bessere „**Separation of Concern**“: Koordinations- und Geschäftslogik
 - + Starke **Entkopplung** der Subsysteme
 - + Bewältigung von **Heterogenität**
 - + Verbesserung der **Erweiterbarkeit**
 - + Verbesserung der **Datenqualität**

- **Nachteile**
 - [Noch] zu wenig **Erfahrungen/Best Practices**
 - Problematische **Performance** bei bestimmten Interface-Operationen
 - Unzureichend genaue **Spezifikation**

- Semantic Spaces
- Behandelte Problemstellungen
 - RQ.1: Koordination mit Semantic Spaces
 - **RQ.2: Effiziente Event-Erkennung**
 - RQ.3: Formale Beschreibung semantischer Event-basierter Systeme
- Evaluierung
- Zusammenfassung

RQ.2: Efficiency of Semantic Event Processing Systems

■ Problemdefinition

- Existierende Semantic Spaces weisen oft mangelhafte Performance auf (blockierende Operationen, Subskriptionen)
- Bsp.:
 - **TripCom Triple Space**: wenige Operationen/Sekunde
 - **SENS**: bis zu 3s bei einer Wissensbasis von 1 Million Triples

RQ.2 Efficiency of Semantic Event Processing Systems.

What optimization methods can be employed to make semantic spaces a suitable platform for the implementation of efficient coordination applications with medium and large-size sets of shared data?

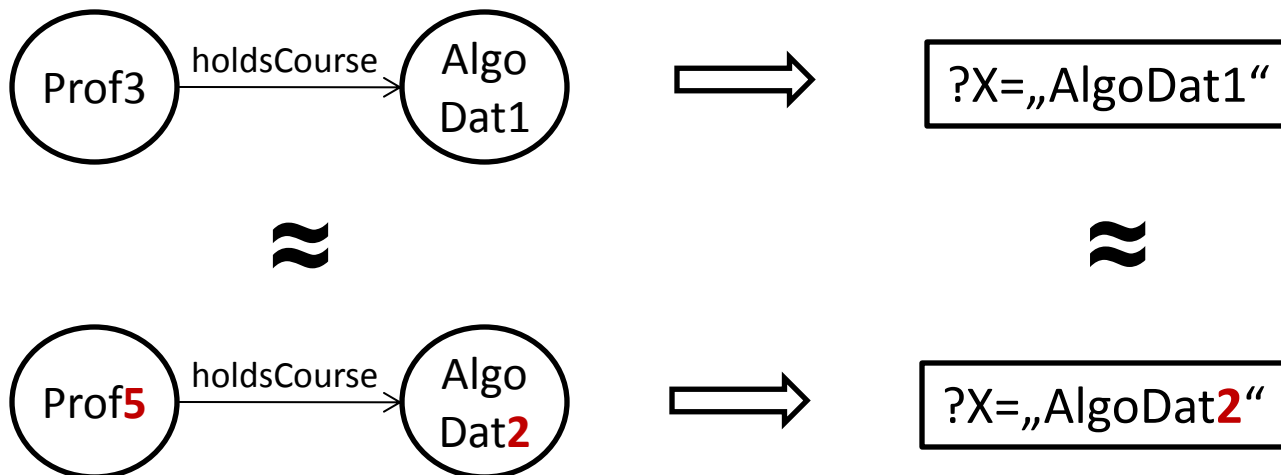
■ Beobachtung

- Die Publikation ähnlicher Graphen führt häufig zu ähnlichen Notifikationen
 - Bsp.: Wenn Informationen über **Personen** publiziert werden, führt dies häufig zu Notifikationen über **Adressänderungen**

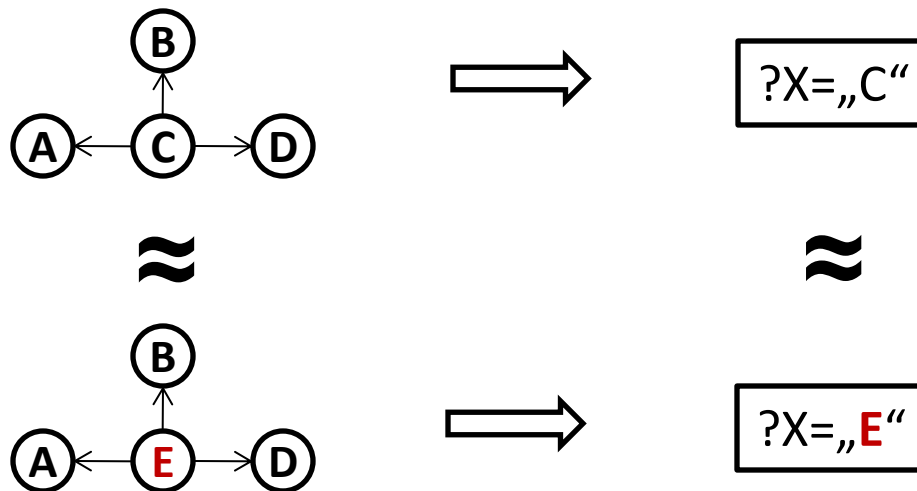
■ Ansatz

- Erkennen solcher Ähnlichkeits-Muster und „erraten“ von Notifikationen
- **Vorteil:** Die Validierung geratener Notifikationen kann wesentlich effizienter ausgeführt werden als die reguläre Evaluierung der Wissensbasis

- Entwicklung eines Heuristik-Frameworks
- Zwei heuristische Methoden
 - 1) **StringRS**: erkennt textuelle Ähnlichkeiten

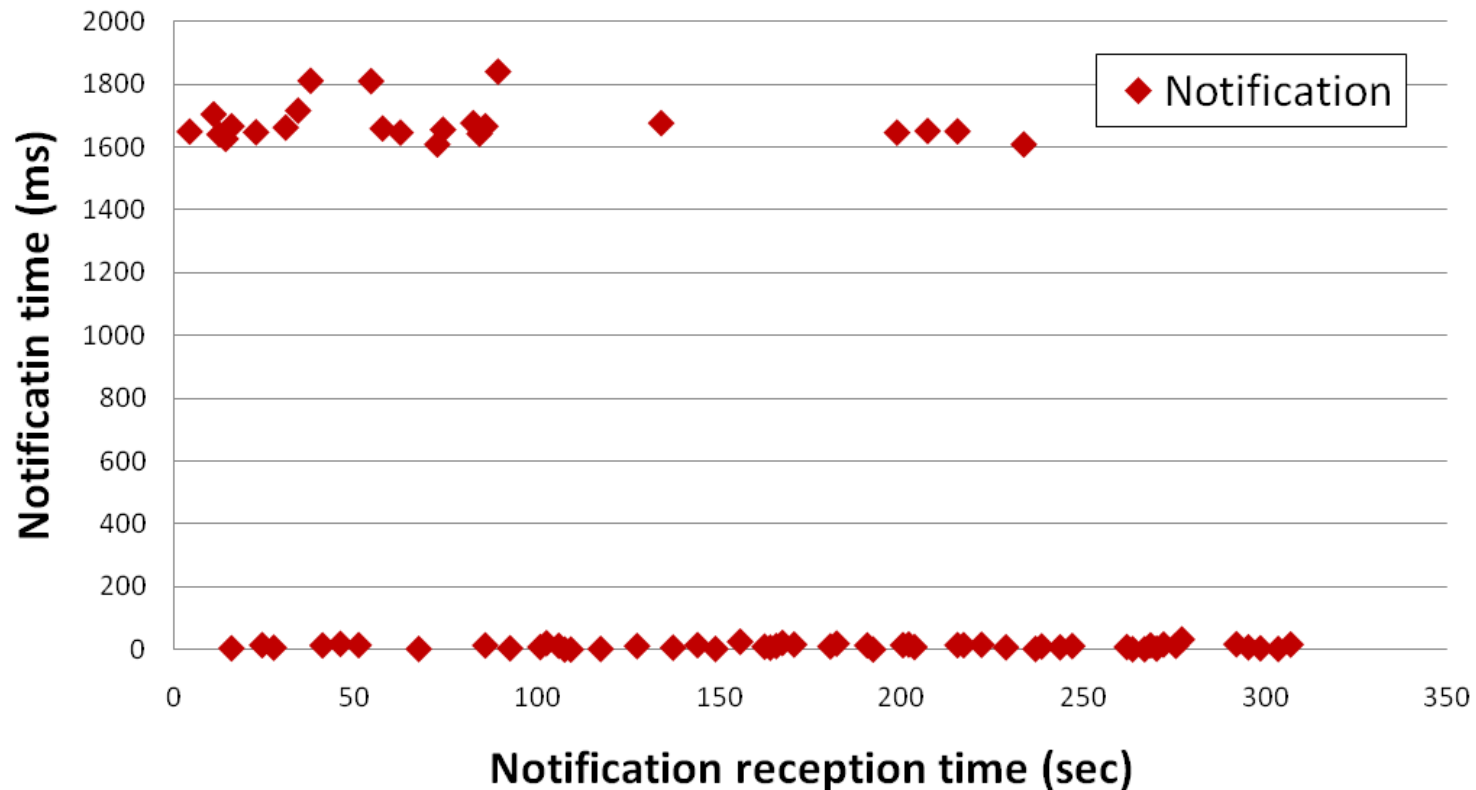


- Entwicklung eines Heuristik-Frameworks
- Zwei heuristische Methoden
 - 2) GraphSS: erkennt strukturelle Ähnlichkeiten**



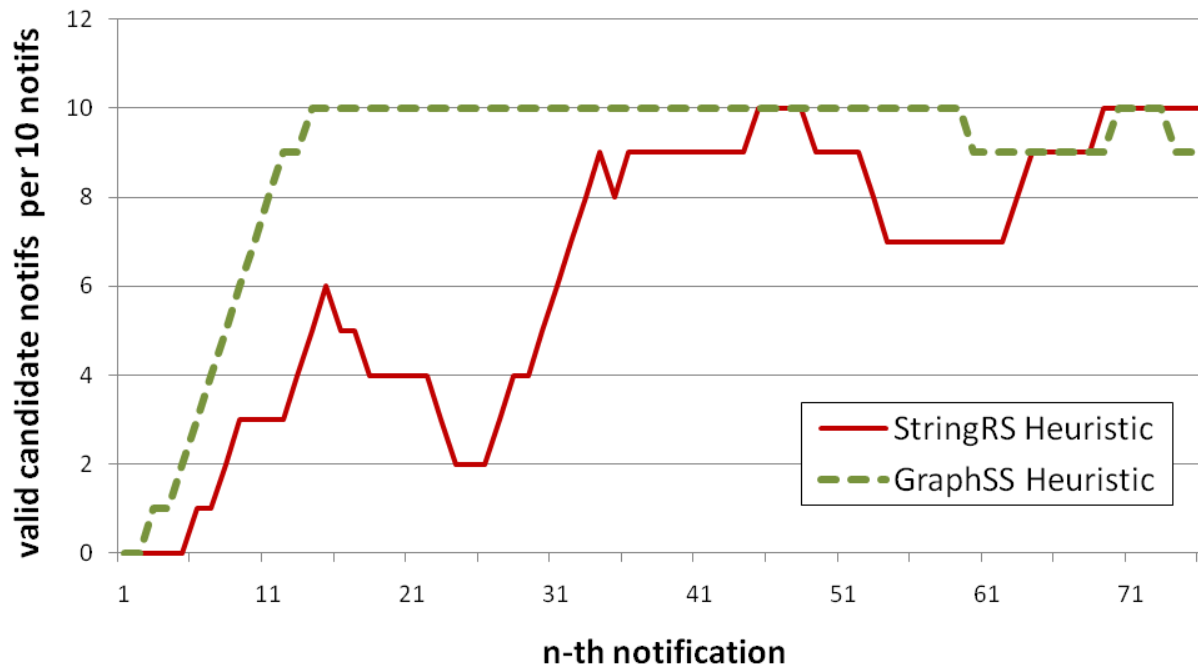
RQ.2: Typisches Verhalten bei heuristischer Event-Erkennung

■ StringRS Heuristik



RQ.2: Optimierung des Durchsatzes

- Bisher wurden nur die Antwortzeiten verbessert
 - Durchsatz bleibt gleich
- Auch ein höherer Durchsatz (Ops/sec) kann erreicht werden
 - Durch Aussetzen der regulären Evaluierung in bestimmten Zeitintervallen



- Semantic Spaces
- Behandelte Problemstellungen
 - RQ.1: Koordination mit Semantic Spaces
 - RQ.2: Effiziente Event-Erkennung
 - **RQ.3: Formale Beschreibung semantischer Event-basierter Systeme**
- Evaluierung
- Zusammenfassung

RQ.3: Formal Description of Semantic Spaces

■ Problemdefinition

- Es existieren keine formalen Methoden um das Verhalten von Semantic Spaces zu spezifizieren
- Bsp.: TripCom
 - Mehrere Revisionen der textuellen Beschreibung

RQ.3 Formal Description of Semantic Spaces.

How can formal concepts be employed for the specification of interface semantics and runtime behavior of semantic space systems?

RQ.3: Ansatz: Formale Spezifikation mit Temporal Propositional Logic (TPL)

- Das **Verhalten** eines Systems wird als **Trace** beschrieben
 - $\sigma = s_0, s_1, s_2, s_3 \dots$
- Jeder **Zustand** eines Traces wird definiert durch das Paar
 - $s_i = (\bar{s}_i, \bar{a}_i)$
 - Bsp.: $s_7 = (D = \{x, y, z\}, \text{notify}(C, q, b))$
- Ein **atomares Predikat P** ist wahr für einen Trace genau dann wenn es für den ersten Zustand des Traces wahr ist.
- Eine **Spezifikation** wird definiert als Menge von erlaubten Traces.

RQ.3: Temporale Formeln

■ Definition von Mengen mit Temporalen Operatoren

- “**always**”: $\Box\varphi$ is true for trace σ iff for all $i \geq 0$, φ is true for the trace $\sigma|_i$.
- ◇ “**eventually**”: $\Diamond\varphi$ is true for trace σ iff there exists an $i \geq 0$, such that φ is true for the trace $\sigma|_i$.
- “**next**”: $\bigcirc\varphi$ is true for trace σ iff φ is true for the trace $\sigma|_1$.
- \mathcal{U} “**until**”: $\varphi \mathcal{U} \psi$ is true for trace σ iff there exists a $j \geq 0$, such that for all $0 \leq i < j$, φ is true for the trace $\sigma|_i$ and ψ is true for the trace $\sigma|_j$.

RQ.3: Definition des Systemzustandes

- Zustandsvariablen eines Semantic Event Processing Systems
 - Gespeicherte Daten D
 - Ableitbares Wissen Z
 - Ableitung unter Verwendung einer Inferenzfunktion $inf: D \rightarrow Z$
 - Aktive Subskriptionen eines Clients S_c
 - Subskriptionen als Queries $q_1 \dots q_n$
 - Jede Query selbst als Abbildung $q_i: Z \rightarrow \{b\}$

RQ.3: Interface-Operationen

- Fünf Interface-Operationen
 - pub, del, sub, unsub, notify
- Änderungen der Zustandsvariablen

Interface Operation	State Changes
$pub(C, K)$	$D' = D \cup K$ $Z' = inf(D')$
$del(C, K)$	$D' = D \setminus K$ $Z' = inf(D')$
$sub(C, q)$	$S'_C = S_C \cup \{q\}$
$unsub(C, q)$	$S'_C = S_C \setminus \{q\}$
$notify(C, q, b)$	-

Def.: Semantic Event Processing System

Definition 2 (Semantic Event Processing System). A Semantic Event Processing System *is a system that exhibits only traces satisfying the following requirements:*

(Safety)

$$\begin{aligned} \square [& notify(C, q, b) \Rightarrow q \in S_C \\ & \wedge b \in q(Z) \\ & \wedge \bigcirc (\neg notify(C, q, b) \mathcal{U} b \notin q(Z))] \end{aligned}$$

- aktive Subskription
- gültiges Ergebnis
- nur 1x notifizieren

(Liveness)

$$\begin{aligned} \square [& \square (q \in S_C) \\ & \Rightarrow [(b \notin q(Z) \wedge \bigcirc (b \in q(Z))) \Rightarrow \bigcirc \Diamond notify(C, q, b)]] \end{aligned}$$

- Auf jede Ableitung eines neuen Ergebnisses muss eine Notifikation folgen

- Semantic Spaces
- Behandelte Problemstellungen
 - RQ.1: Koordination mit Semantic Spaces
 - RQ.2: Effiziente Event-Erkennung
 - RQ.3: Formale Beschreibung semantischer Event-basierter Systeme
- **Evaluierung**
- Zusammenfassung

- **RQ.1: „Coordination with Semantic Spaces“**
 - Recherche und Sammlung von bereits evaluierten Einzelarbeiten
 - Implementierung und Anwendung der Konzepte
- **RQ.2: „Efficiency of Semantic Event Processing Systems“**
 - Evaluierung anhand eines eigens entwickelten Benchmark Frameworks für Semantic Event Processing Systeme ...→
- **RQ.3: „Formal Description of Semantic Spaces“**
 - Reviews der formalen Konzepte durch Konferenzpublikation
 - Anwendung der entwickelten Konzepte zur Spezifikation von SENS

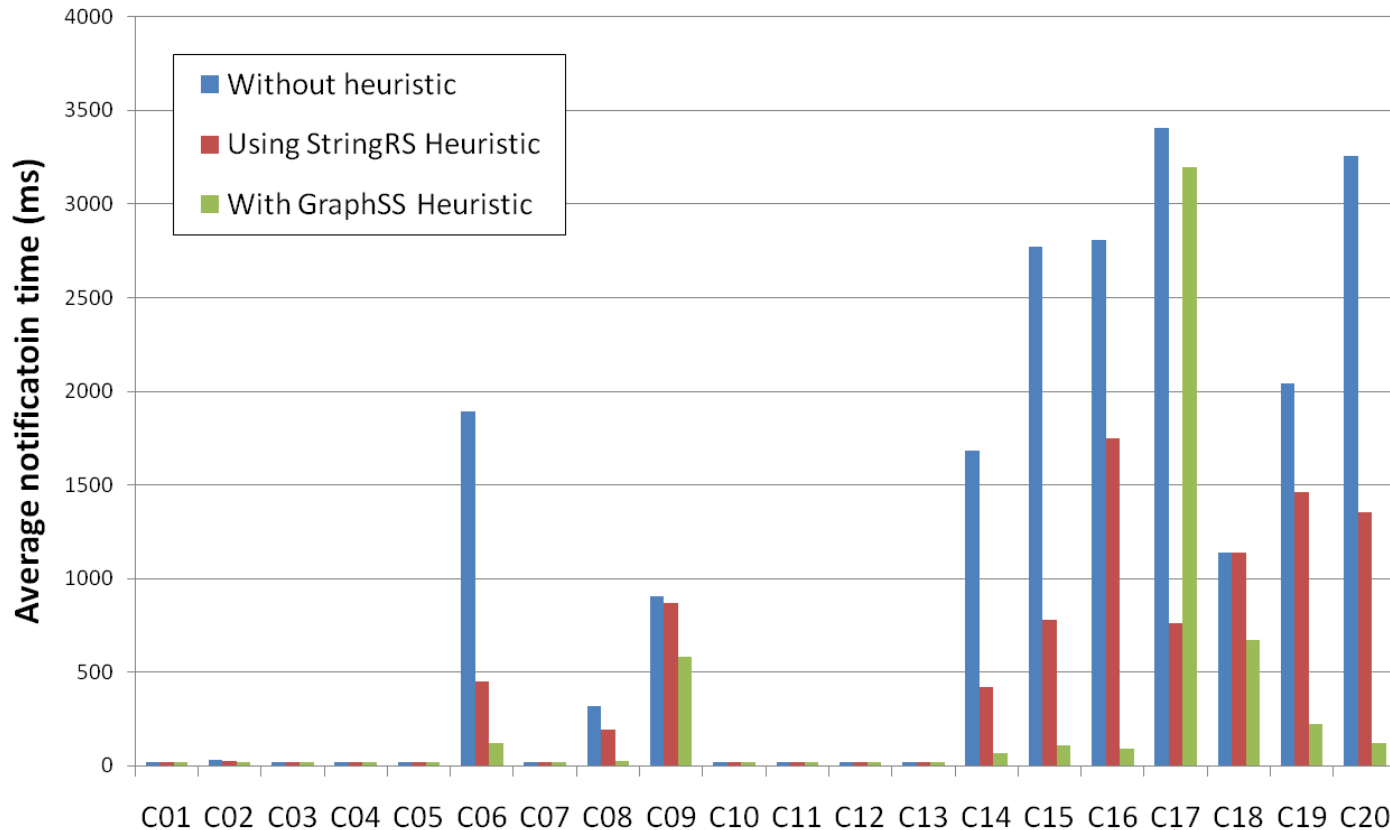
■ Benchmark

- Basierend auf LUBM Benchmark für Semantic Databases
- Sequenz von 300 Publikationsoperationen
- 20 verschiedene Subskriptionen (Channels)
- Bei einer Wissenbasis von 2,4 Mio. Triples

■ Metriken

- Notifikationszeit
 - Zeit zwischen dem frühestmöglichen und dem tatsächlichen Erkennen eines Events
 - Kenngröße für das Antwortzeitverhalten
- Publikationsdurchsatz
 - Anzahl der abgearbeiteten Publikationsoperationen pro Zeiteinheit
 - Kenngröße für die maximale Gesamtlast

Durchschnittliche Notifikationszeiten

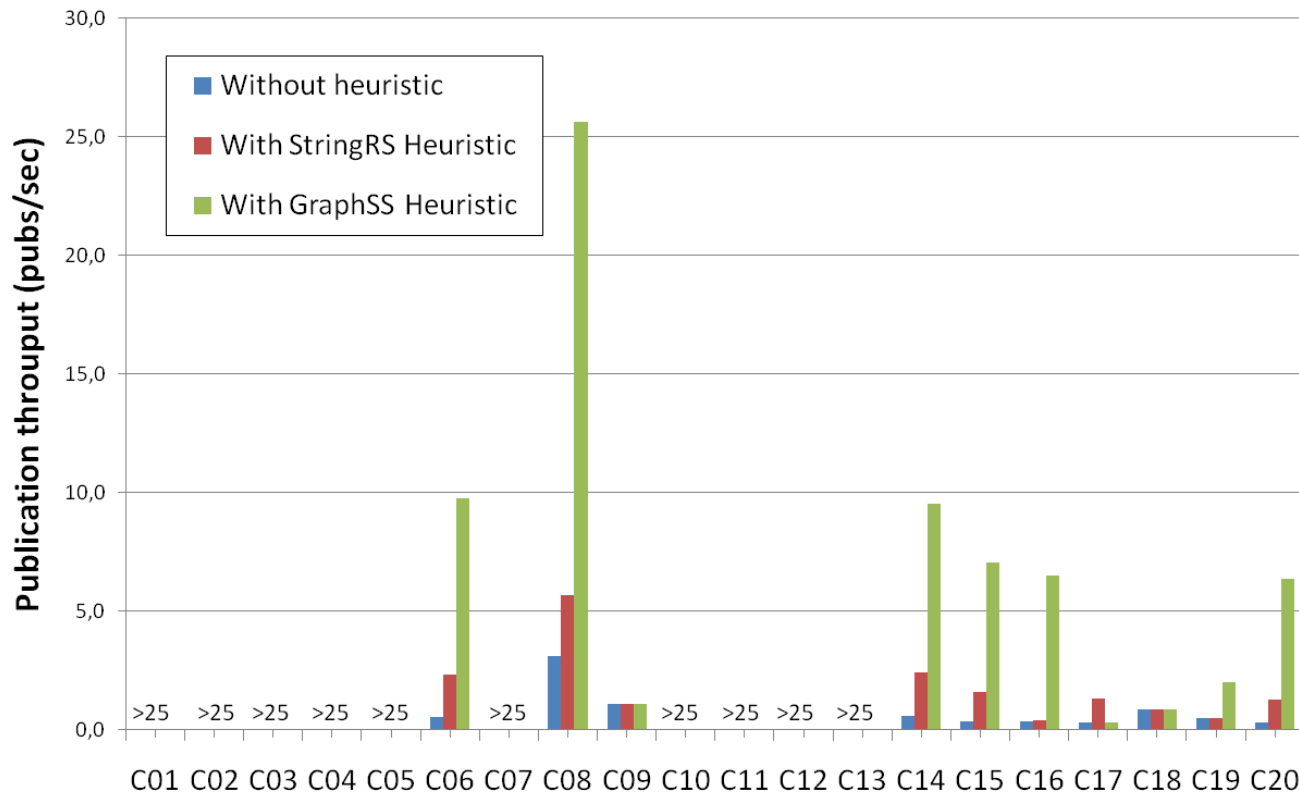


Verbesserung um 78%

Verbesserung um 97%



■ Durchschnittliche Publikationsdurchsatz-Raten



Verbesserung um Faktor 4,4

Verbesserung um Faktor 20,5



- **RQ.1 „Coordination with Semantic Spaces“**
 - + **Stärken:** Ausdruckskraft, Separation of Concern, Entkopplung, Heterogenität, Erweiterbarkeit, Datenqualität
 - **Schwächen:** Performance, Unzureichende Spezifikationen
- **RQ.2 „Efficiency of Semantic Event Processing Systems“**
 - Verbesserung der Performance mit **Heuristik-Framework**
 - **Essentielle Verbesserungen** für viele aber nicht alle Testfälle
- **RQ.3 „Formal Description of Semantic Spaces“**
 - Verwendung von **Temporal Logic**
 - **Präzise und verständliche** Spezifikation von Semantic Spaces

- Die entwickelten Optimierungsmethoden ...
 - setzen die Verwendung von „**Description Logics**“ und „**Open World Assumption**“ voraus

- Weiterführende Forschungsfragen:
 - Können die entwickelten Methoden auch mit anderen/erweiterten Inferenzmechanismen verwendet werden?
 - *Bsp.:* Erweiterung um Regel-basiertes Reasoning

 - Welche Auswirkungen haben diese Erweiterungen auf die Schnittstellensemantik und das Laufzeitverhalten von Semantic Spaces?
 - *Bsp.:* Erweiterung des formalen Modells

RQ.3: Formale Eigenschaften von Semantic Spaces

- **Reliable Notification Delivery**
 - Client wird über jede einzelne neue Ableitung eines Binding Sets benachrichtigt
- **Channel-wide Notification Sequence**
 - Alle Subscriber der gleichen Subskription erhalten Notifikationen in der gleichen Reihenfolge
- **Causal Notification Ordering**
 - Die Reihenfolge von Notifikationen die in einem kausalen Zusammenhang stehen wird zuverlässig eingehalten
- **Inference Time Notification Ordering**
 - Clients werden immer in der Reihenfolge benachrichtigt, in der neue Ergebnisse im Semantic Space abgeleitet werden können

RQ.3: Von SENS gewährleistete Eigenschaften

- Ohne bzw. mit heuristischer Optimierung

Condition	SENS without opt.	SENS heuristic opt.
Safety (required)	✓	✓
Liveness (required)	✓	✓
Reliable Notification Delivery	✓	✓
Channel-wide Notification Sequence	✓	✓
Causal Notification Ordering	✓	✓
Inference Time Notification Ordering	✓	×

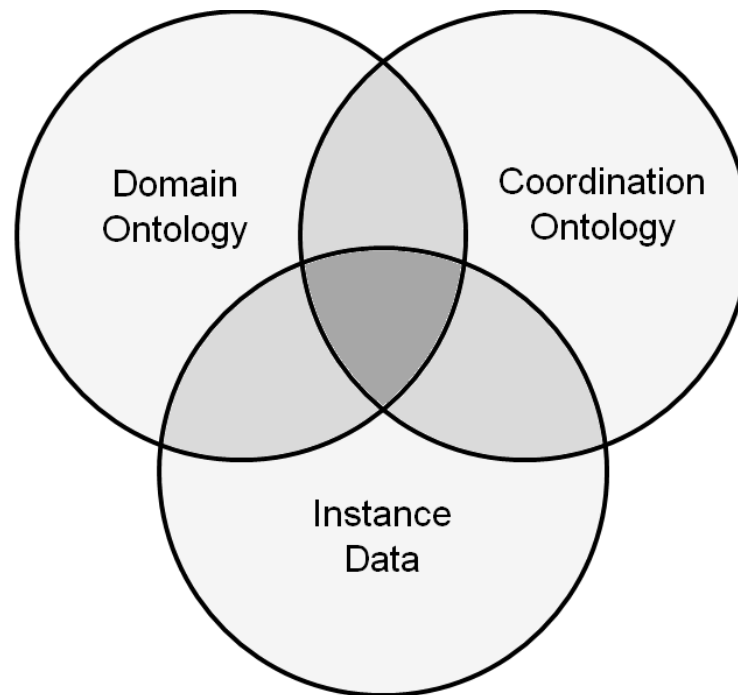
- Referenzdatengenerierung
- LUBM
 - LUBM Data Generator generiert synthetische Daten
 - Zu homogen für aufschlussreiche Tests
- XLUBM Benchmark Framework
 - Entwickelt für das Benchmarking von Semantic Event Processing Systems
 - Generierung von realistischen RDF-Daten
 - Extraktion von Daten aus DBpedia, z.B.:
 - „Prof1“ → „Heinz Zemanek“
 - „PaperTitle2“ → „Coordination with Semantic Spaces“

Benchmark Ergebnisse / Beobachtungen

- Effiziente Auswertung auch ohne Optimierung wenn ...
 - einfaches oder kein Reasoning notwendig
 - Beobachtung konkreter Ressourcen
 - Abfragen liefern kleine Ergebnismengen

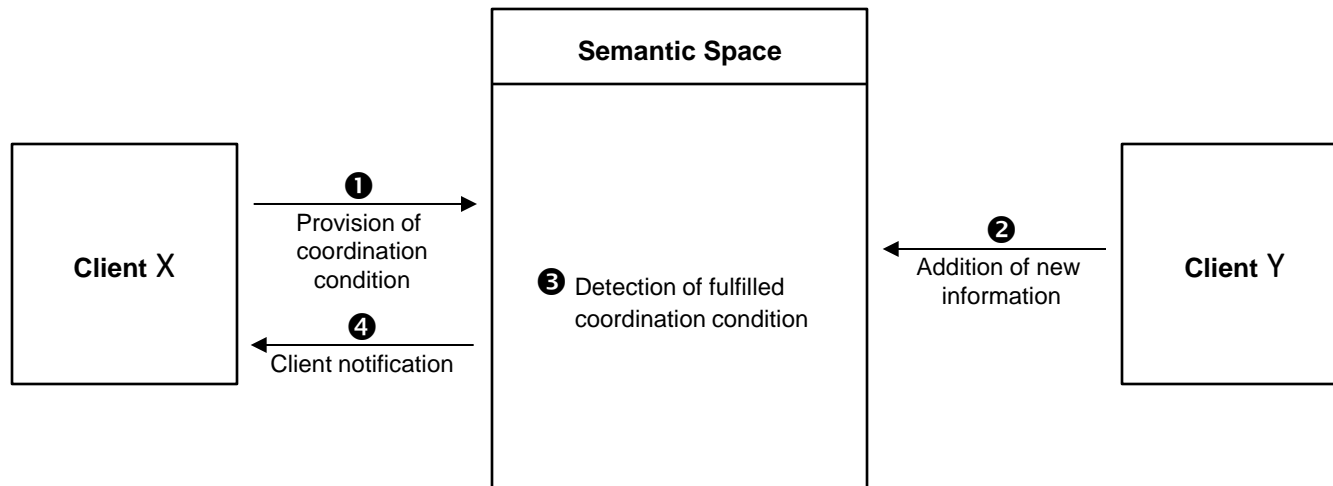
- Effiziente Auswertung mit Heuristiken wenn ...
 - Subskriptionen generieren viele Notifikationen
 - geringe Zahl an Variablen
 - **StringRS**: viele gleiche Textblöcke in Graphen und Notifikationen
 - **StringRS**: alle Variablenwerte neuer Notifikationen im selben Graph
 - **GraphSS**: ähnliche Größe der publizierten Graphen
 - **GraphSS**: „zentraler Knoten“ von Graphen die zu ähnlichen Notifikationen führen vom gleichen Typ

Strukturmuster: Fragmente der Wissensbasis



Verhaltensmuster:

Implementierung eines Koordinationsschrittes



Graph Pattern of Subscription “Q2”⁵

prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

prefix uni : <http://www.uni.org/onto#>

?L rdf:type uni:Lecturer *(subject, predicate, object)*

?L uni:holdsCourse ?C *(subject, predicate, object)*

?C uni:name ?N *(subject, predicate, object)*

GraphSS Heuristik: Publikation und Notifikation (1)

Published Graph

```
prefix foaf : <http://xmlns.com/foaf/0.1/>
prefix uni  : <http://www.uni.org/onto#>
prefix uld1 : <http://www.unil.org/depl/>

uld1:persons/Professor137 foaf:firstName "Heinz"
uld1:persons/Professor137 foaf:surname "Zemanek"
uld1:persons/Professor137 uni:teaches uld1:courses/Course148
uld1:courses/Course148 uni:name "History of Computer Science"
```

Graph Similarity Class = {foaf:firstName, foaf:surname, uni:teaches}

Notification for Subscription "S2"

```
?L = http://www.unil.org/depl/persons/Professor137
?C = http://www.unil.org/depl/courses/148
?N = "History of Computer Science"
```

Notification Similarity Class = ("Q2")

GraphSS Heuristik: Similarity Classes und Mapping

Graph Similarity Class	Notification Similarity Class	Mapping Attributes
...
{foaf:firstName, foaf:surname, uni:teaches}	S2	BVPDs={ (?L, foaf:firstName, <i>predicate</i> , <i>subject</i>), (?C, uni:teaches, <i>predicate</i> , <i>subject</i>), (?N, uni:name, <i>predicate</i> , <i>object</i>)}

GraphSS Heuristik: Publikation (2) und generierte Notifikation

Published Graph

```
prefix foaf : <http://xmlns.com/foaf/0.1/>
prefix uni  : <http://www.uni.org/onto#>
prefix u2d1 : <http://www.uni2.org/depl/>

u2d1:persons/Professor102 foaf:firstName "Niklaus"
u2d1:persons/Professor102 foaf:surname "Wirth"
u2d1:persons/Professor102 uni:teaches u2d1:courses/Course252
u2d1:courses/Course252 uni:name "Programming in Oberon"
u2d1:courses/Course252 uni:ects 4
```

Graph Similarity Class = {foaf:firstName, foaf:surname, uni:teaches}

Candidate Notification for Subscription "Q2"

```
?L = http://www.unil.org/depl/persons/Professor102
?C = http://www.unil.org/depl/courses/252
?N = "Programming in Oberon"
```

RQ.2 (Efficiency) Verwandte Ansätze

- **Semantische Datenbanken**
 - Optimiert für „**One-time Reasoning**“ und „**One-time Querying**“
 - In Semantic Spaces: **Häufig wiederholtes** Reasoning und Querying auf sich verändernden Wissensbasen
- **CEP – Complex Event Processing**
 - Erkennung von komplexen Mustern auf **Event-Streams**
 - In Semantic Spaces: Erkennung von komplexen Mustern sich ändernder **Information**
- **Aktive Datenbanken**
 - Benachrichtigung über **explizit** hinzugefügte Daten
 - In Semantic Spaces: Benachrichtigung über **implizit** ableitbare Daten

RQ.3 (Formalism): Verwandte Ansätze

- **Modellierung und Spezifikation von Verteilten Systemen**
 - Prozessalgebren bzw. –kalkuli (zB. Busi, Gorrieri, Zavattaro für Linda)
 - Petri-Nets
 - Temporale Logik
- **Vorteil von Temporaler Logik**
 - Einfache Abbildung eines Zustands mithilfe von klassischer Logik (Aussagenlogik, Prädikatenlogik)
 - Vorarbeiten auf dem Gebiet der Spezifikation von verteilten Event-basierten Systemen