A Basic Calculus for Modeling Service Level Agreement

Rocco De Nicola **Rosario Pugliese**
{denicola,pugliese}@dsi.unifi.it

Dipartimento di Sistemi e Informatica,
Università di Firenze,

Gianluigi Ferrari **Ugo Montanari** **Emilio Tuosto**
{giangi,ugo,etuosto}@di.unipi.it

Dipartimento di Informatica,
Università di Pisa

COORDINATION05: Namur, 20 – 23 April 2005

Profundis

Overview of the talk

“Ayudadme a comprender lo que os digo y os lo explicaré mejor”

“Help me in understanding what I’m saying and I will explain it better”

(Antonio Machado)

Overview

- A few motivations
- Background: constraint semirings (c-semirings)
- $\mathcal{K}oS$
  - design choices...
  - ...weighted links
- Syntax and semantics of $\mathcal{K}oS$
- Examples
- Conclusion

Motivations

The real technology - behind all of our technologies - is language

(N. Fisher)

Global Computing and Services

- Service Oriented Computing
  - applications are made by gluing *services*
    - “autonomous”
    - independent (local choices, independently built)
    - mobile/stationary
    - “interconnected”
  - interactions governed by programmable coordination policies
  - services are searched and binded ... offline

Global Computing and Services

- Service Oriented Computing
  - applications are made by gluing *services*
    - “autonomous”
    - independent (local choices, independently built)
    - mobile/stationary
    - “interconnected”
  - interactions governed by programmable coordination policies
  - services are searched and binded ... offline

Can search/bind be dynamic and at run-time?

Global Computing and Services

- Service Oriented Computing
  - applications are made by gluing *services*
    - “autonomous”
    - independent (local choices, independently built)
    - mobile/stationary
    - “interconnected”
  - interactions governed by programmable coordination policies
  - services are searched and binded ... offline

Can search/bind be dynamic and at run-time?

- Search and bind wrt application level QoS
  - not low-level performance (e.g., throughput, response time)
  - but application-related, e.g.
    - price services
    - payment mode
    - data available in a given format

Our approach in brief

WAN programming is not just **go**(P), $\bar{s}\langle x \rangle$ or $s(y)$

- Lifting QoS issues to application level...
- ...for programming global computers
- with programmable application level QoS
- and develop proof techniques and tools

First steps (extending Klaim) in [DFM+03]

We are currently distilling Klaim into $\mathcal{K}o\mathcal{S}$ which exploits c-semiring for

- expressing application level QoS dependent connections and
- for coordinating remote activities...
- ...by means of c-semiring values

Our approach in brief

WAN programming is not just **go**(P), $\bar{s}\langle x \rangle$ or $s(y)$

- Lifting QoS issues to application level...
- ...for programming global computers
- with programmable application level QoS
- and develop proof techniques and tools

First steps (extending Klaim) in [DFM+03]

We are currently distilling Klaim into $\mathcal{K}oS$ which exploits c-semiring for

- expressing application level QoS dependent connections and
- for coordinating remote activities...
- ...by means of c-semiring values

We also are defining a graph-based model [HT05]

Our approach in brief

WAN programming is not just **go**(P), $\bar{s}\langle x \rangle$ or $s(y)$

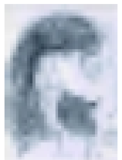
- Lifting QoS issues to application level...
- ...for programming global computers
- with programmable application level QoS
- and develop proof techniques and tools

First steps (extending Klaim) in [DFM+03]

We are currently distilling Klaim into $\mathcal{K}o\mathcal{S}$ which exploits c-semiring for

- expressing application level QoS dependent connections and
- for coordinating remote activities...
- ...by means of c-semiring values

We also are defining a graph-based model [HT05]

Ask me at the poster session...look for  or ... 

Background

"During my nine years at the elementary schools
I was not able to teach anything to my professors"

(Bertolt Brecht)

Constraint Semirings

C-Semirings [BMR95, BMR97] for abstracting application level QoS

- $\langle A, +, \star, \mathbf{0}, \mathbf{1} \rangle$, where
  - A is a set (containing $\mathbf{0}$ and $\mathbf{1}$),
  - $+, \star : A \times A \rightarrow A$

$+$	$\star$
$x + y = y + x$	$x \star y = y \star x$
$(x + y) + z = x + (y + z)$	$(x \star y) \star z = x \star (y \star z)$
$x + \mathbf{0} = x$	$x \star \mathbf{0} = \mathbf{0}$
$x + \mathbf{1} = \mathbf{1}$	$x \star \mathbf{1} = x$
$x + x = x$	$(x + y) \star z = (x \star z) + (y \star z)$

Constraint Semirings

C-Semirings [BMR95, BMR97] for abstracting application level QoS

- $\langle A, +, \star, \mathbf{0}, \mathbf{1} \rangle$, where
  - A is a set (containing $\mathbf{0}$ and $\mathbf{1}$),
  - $+, \star : A \times A \rightarrow A$

$+$	$\star$
$x + y = y + x$	$x \star y = y \star x$
$(x + y) + z = x + (y + z)$	$(x \star y) \star z = x \star (y \star z)$
$x + \mathbf{0} = x$	$x \star \mathbf{0} = \mathbf{0}$
$x + \mathbf{1} = \mathbf{1}$	$x \star \mathbf{1} = x$
$x + x = x$	$(x + y) \star z = (x \star z) + (y \star z)$

C-Semirings [BMR95, BMR97] for abstracting application level QoS

- $\langle A, +, \star, \mathbf{0}, \mathbf{1} \rangle$, where
  - A is a set (containing $\mathbf{0}$ and $\mathbf{1}$),
  - $+, \star : A \times A \to A$

$+$	$\star$
$x + y = y + x$	$x \star y = y \star x$
$(x + y) + z = x + (y + z)$	$(x \star y) \star z = x \star (y \star z)$
$x + \mathbf{0} = x$	$x \star \mathbf{0} = \mathbf{0}$
$x + \mathbf{1} = \mathbf{1}$	$x \star \mathbf{1} = x$
$x + x = x$	$(x + y) \star z = (x \star z) + (y \star z)$

- Implicit partial order: $a \leq b \iff a + b = b$ "b is *better* than a"

A bunch of c-semirings

C-semirings structures can been defined for many frameworks:

- $\langle \{\mathit{true}, \mathit{false}\}, \vee, \wedge, \mathit{false}, \mathit{true} \rangle$ (boolean): Availability
- $\langle \mathrm{Real}+, \mathit{min}, +, +\infty, 0 \rangle$ (optimization): Price, propagation delay
- $\langle \mathrm{Real}+, \mathit{max}, \mathit{min}, 0, +\infty \rangle$ (max/min): Bandwidth
- $\langle [0,1], \mathit{max}, \cdot, 0, 1 \rangle$ (probabilistic): Performance and rates
- $\langle [0,1], \mathit{max}, \mathit{min}, 0, 1 \rangle$ (fuzzy): Performance and rates
- $\langle 2^N, \cup, \cap, \emptyset, N \rangle$ (set-based, where N is a set): Capabilities

Proposition 1 *The cartesian product of c-semirings is a c-semiring.*

KoS

Ὀμνύω Ἀπόλλωνα ἰητρὸν καὶ Ἀσκληπιὸν καὶ Ὑγείαν καὶ Πανάκειαν καὶ θεοὺς πάντας ἵστορας ποιεύμενος ἐπιτελέα ποιήσειν κατὰ δύναμιν καὶ κρίσιν ἐμὴν ὅρκον τόνδε καὶ ξυγγραφὴν τήνδε ....

διαιτήμασι τε χρήσομαι ἐπ' ὠφελείῃ καμνόντων κατὰ δύναμιν καὶ κρίσιν ἐμὴν ἐπὶ δηλήσει δὲ καὶ ἀδικίῃ εἴρξειν.

οὐ δώσω δὲ οὐδὲ φάρμακον οὐδενὶ αἰτηθεὶς θανάσιμον οὐδὲ ὑφηγήσομαι ξυμβουλίην τοιήνδε· ὁμοίως δὲ οὐδὲ γυναικὶ πεσσὸν φθόριον δώσω.

ἁγνῶς δὲ καὶ ὁσίως διατηρήσω βίον ἐμὸν καὶ τέχνην ἐμήν.

οὐ τεμέω δὲ οὐδὲ μὴν λιθιῶντας, ἐκχωρήσω δὲ ἐργάτῃσιν ἀνδράσιν πρήξιος τῆσδε.

ἐς οἰκίας δὲ ὁκόσας ἂν ἐσίω, ἐσελεύσομαι ἐπ' ὠφελείῃ καμνόντων ἐκτὸς ἐὼν πάσης ἀδικίης ἑκουσίης καὶ φθορίης τῆς τε ἄλλης καὶ ἀφροδισίων ἔργων ἐπί τε γυναικείων σωμάτων καὶ ἀνδρείων ἐλευθέρων τε καὶ δούλων.

ἃ δ' ἂν ἐν θεραπείῃ ἢ ἴδω ἢ ἀκούσω ἢ καὶ ἄνευ θεραπηίης κατὰ βίον ἀνθρώπων, ἃ μὴ χρή ποτε ἐκλαλέεσθαι ἔξω, σιγήσομαι ἄρρητα ἡγεύμενος εἶναι τὰ τοιαῦτα.

ὅρκον μὲν οὖν μοι τόνδε ἐπιτελέα ποιέοντι καὶ μὴ ξυγχέοντι εἴη ἐπαύρασθαι καὶ βίου καὶ τέχνης δοξαζομένῳ παρὰ πᾶσιν ἀνθρώποις ἐς τὸν αἰεὶ χρόνον, παραβαίνοντι δὲ καὶ ἐπιορκοῦντι τἀναντία τουτέων.

Oath of Hippocrates

$\mathcal{K}oS$ characteristics

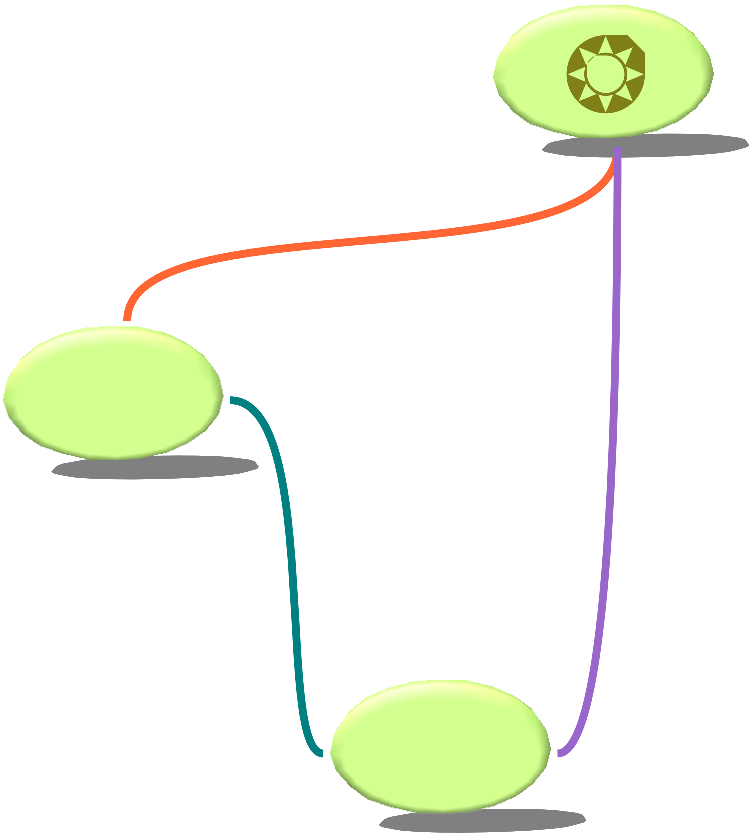
$\mathcal{K}oS$ aims at being a minimal calculus for SOC

- $\mathcal{K}oS$ builds on Klaim (e.g., processes are localised)
- 'cause it naturally supports a *peer-to-peer* programming model
- $\mathcal{K}oS$ primitives handle QoS values as first class entities
- $\mathcal{K}oS$ semantics ensures that the QoS values are respected during execution

- only local communications (unlike Klaim)
- link construction primitives
- links are "mono-use"
- only one remote action
- which relies on link topology
- semantic transitions report the "cost" of the execution

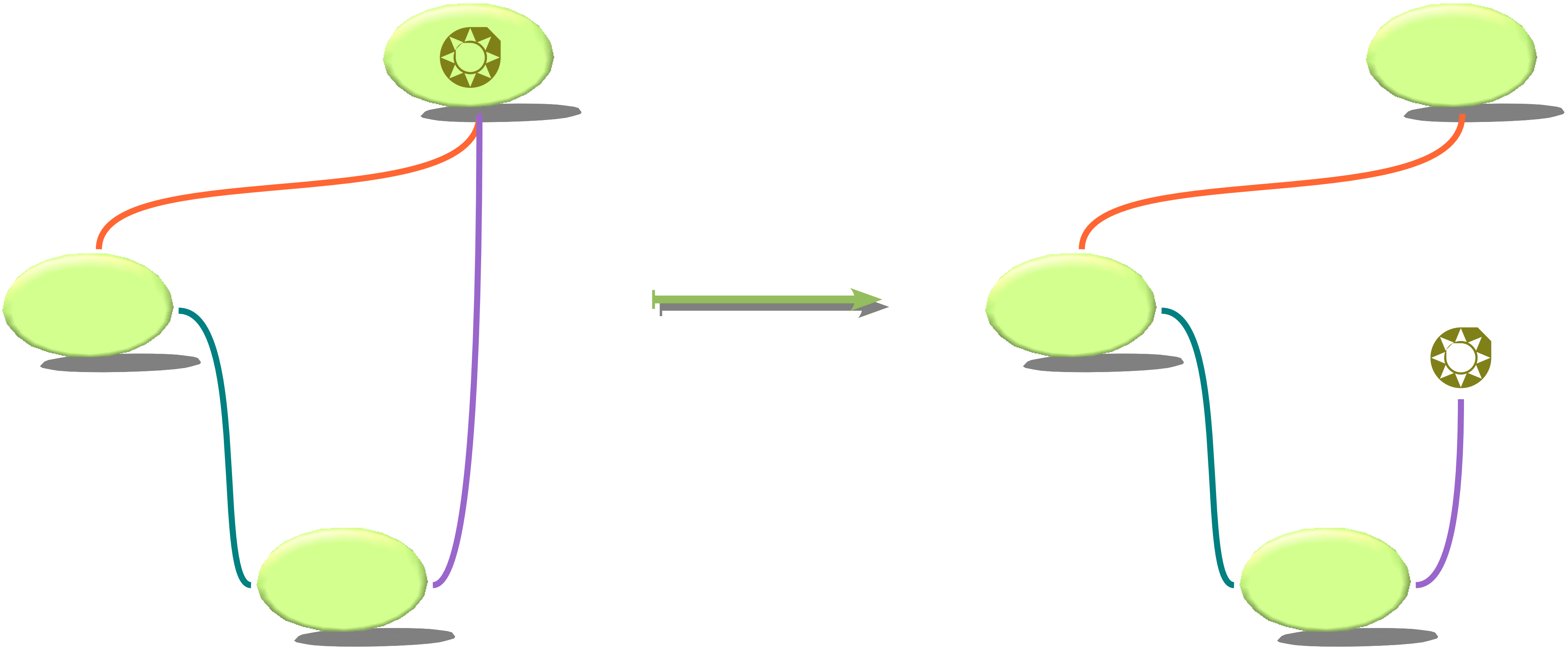
$\mathcal{K}oS$ ... graphically

$\mathcal{K}o\mathcal{S}$ ... graphically

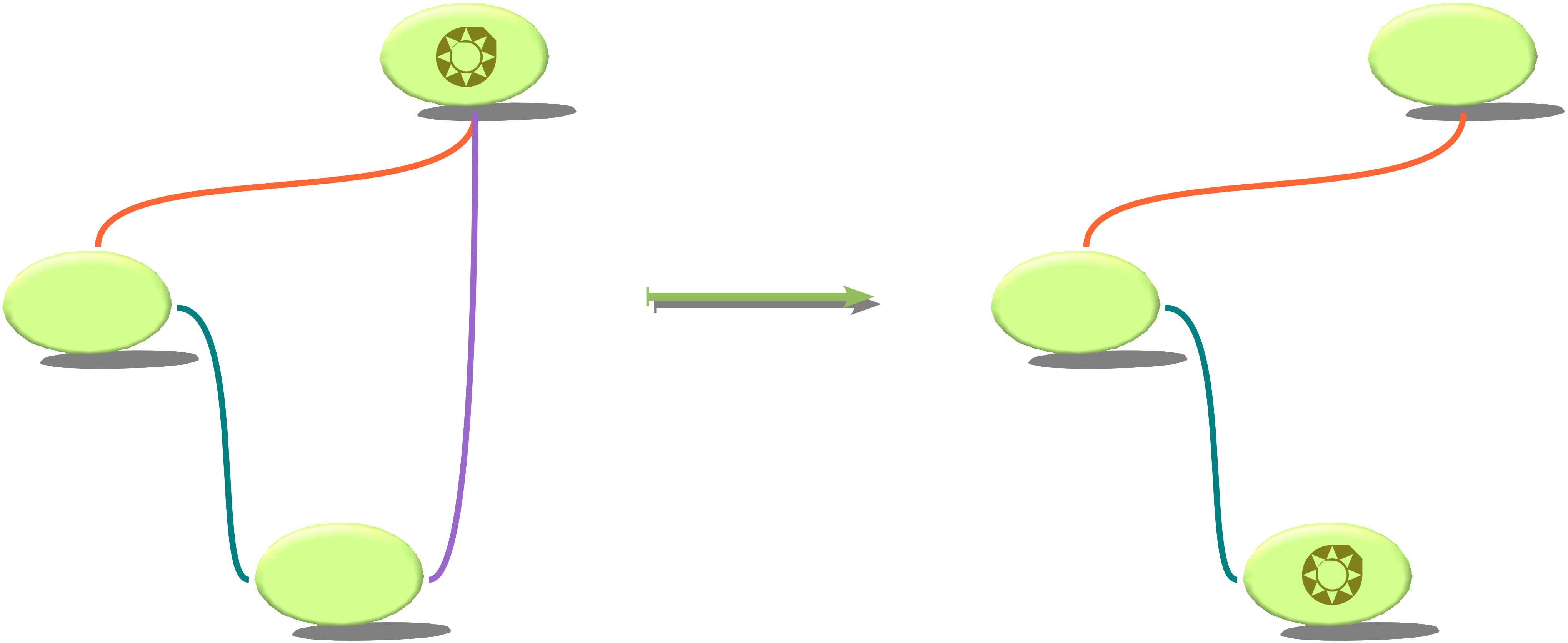


Migration in $\mathcal{K}oS$

$\mathcal{K}oS$ ... graphically

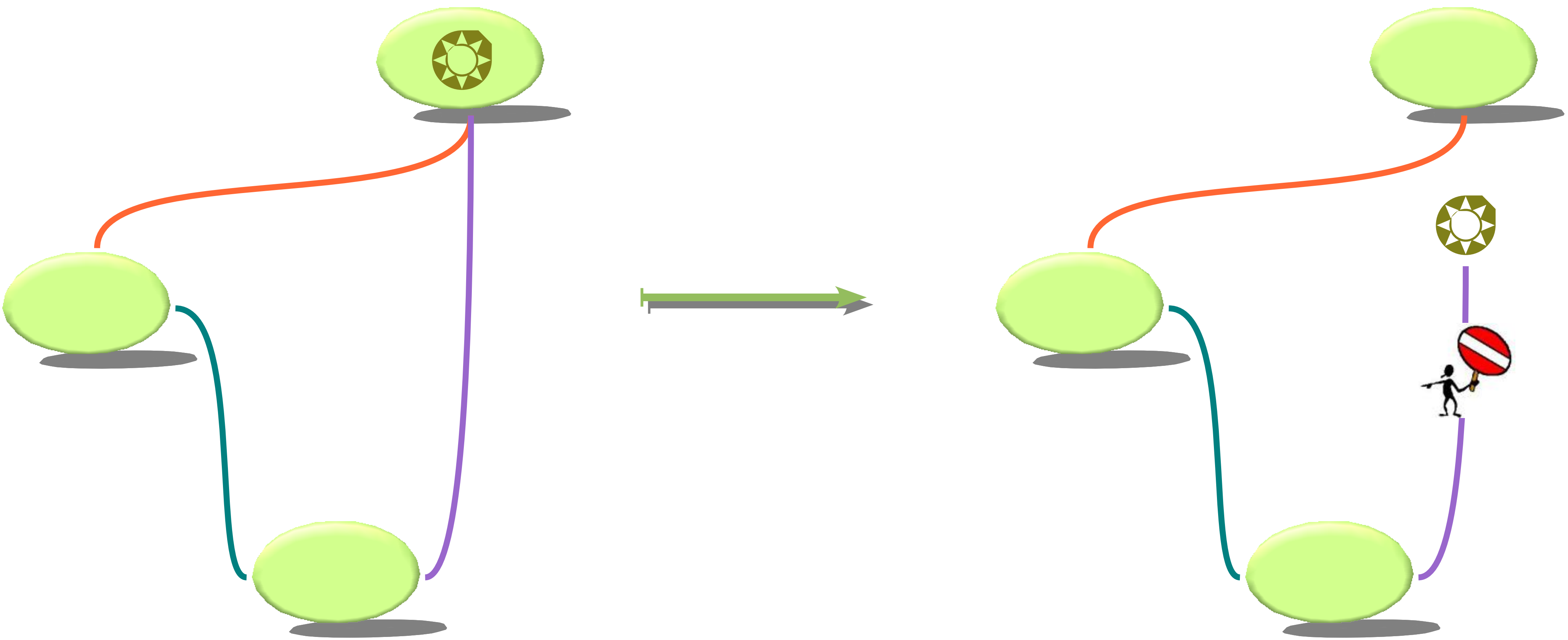


$\mathcal{K}oS$ ... graphically



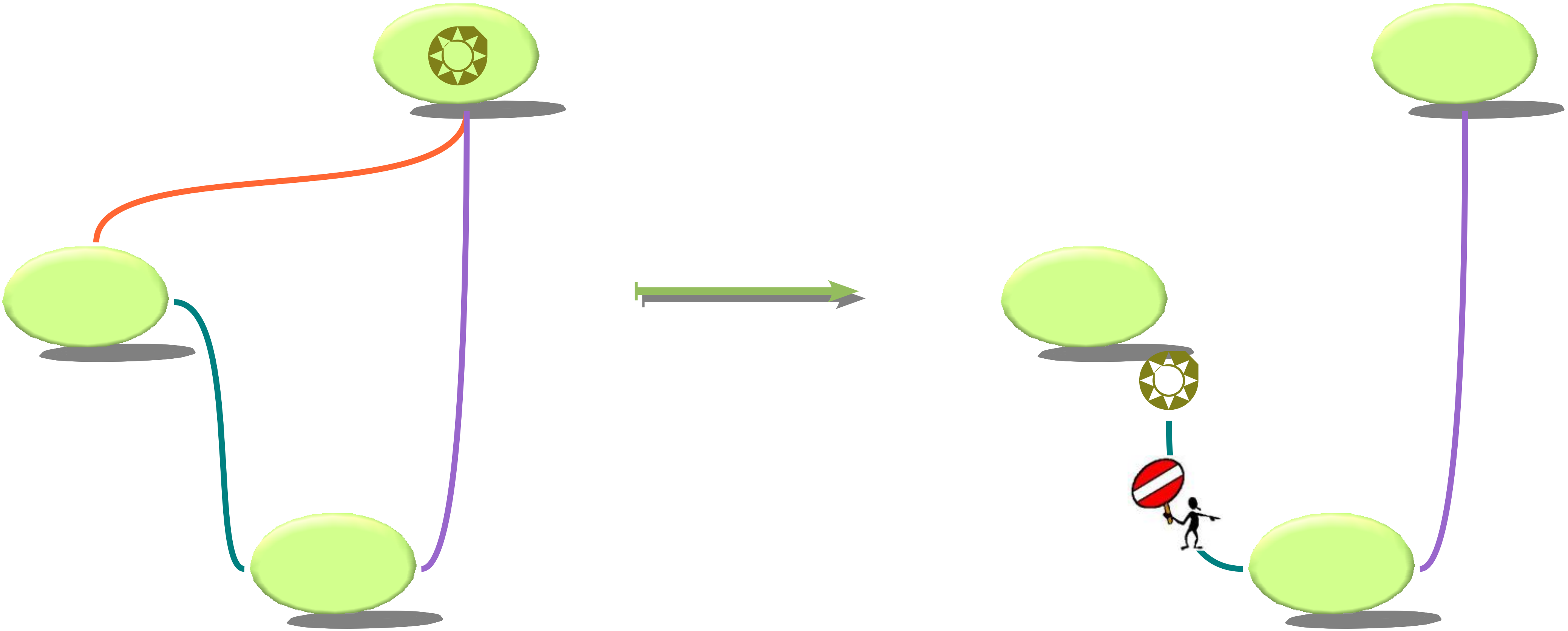
Migration in $\mathcal{K}o\mathcal{S}$

$\mathcal{K}o\mathcal{S}$ ... graphically



Migration in $\mathcal{K}o\mathcal{S}$

$\mathcal{K}o\mathcal{S}$ ... graphically



A motivating example

Consider a scenario where n servers provide services to m clients and focus on balancing the load of the servers.

- clients (c_i) and servers (s_j) are located on different nodes
- c_i issues requests to s_j by spawning a process R

A generic client is described by the following term:

$$c_i :: \langle s_1, \kappa_1 \rangle \quad | \quad \ldots \quad | \quad \langle s_n, \kappa_n \rangle \quad | \quad !C_\delta$$

- $\langle s_j, \kappa_j \rangle$ represents the load κ_j of the server s_j perceived b c_i
- C_δ and R specify the behaviour of c_i:

$$\begin{array}{rcl} C_\delta & \triangleq & (?u, ?v).\varepsilon_v[R]@u.con_{v \star \delta}\langle u \rangle.\langle u, v \star \delta \rangle \\ R & \triangleq & (?x).\langle x+1 \rangle \ldots actual\ request \ldots (?y).\langle y-1 \rangle \end{array}$$

Remark 1 *Remote spawning consumes the traversed links, hence c_i attempts to re-establish a connection with the server!*

A motivating example[2]

s_j is described as:

$$s_j :: \langle h \rangle \;|\; \langle c_1, \kappa'_1 \rangle \;|\; \ldots \;|\; \langle c_m, \kappa'_m \rangle \;|\; !(S\ c_1\ s_j) \;|\; \ldots \;|\; !(S\ c_m\ s_j)$$

- $\langle c_i, \kappa'_i \rangle$ records the QoS value κ'_i assigned to the link towards c_i
- $\langle h \rangle$ is the current load of the s_j
- $S\ c_i\ s_j$ is a *load manager* for c_i

$$\begin{aligned} S\ c\ s \;\triangleq\; & (?l).\langle l \rangle.\mathit{If}_s\ l < max \\ & \quad then\ (c, ?v).acc_{f(v,l)}\langle c \rangle.\langle c, f(v,l) \rangle. \end{aligned}$$

S repeatedly acquires $\langle h \rangle$ and depending on the load decides whether to accept requests for new connections coming from c.

$\mathcal{K}o\mathcal{S}$ syntax

Let $\mathcal{C}$ be the c-semiring of QoS values (ranged over by κ)

$$
\begin{array}{lll}
N, M & ::= & \\
& & \mathbf{0} \\
& | & s :: P \\
& | & (\nu\, s)N \\
& | & N \parallel M \\
& | & s \overset{\kappa}{\frown} t
\end{array}
$$

$$
\begin{array}{lll}
\gamma & ::= & \\
& & (T) \\
& | & \langle v_1, \ldots, v_n \rangle \\
& | & \varepsilon_{\kappa}[P]@t \\
& | & node_{\kappa}\langle t \rangle \\
& | & con_{\kappa}\langle t \rangle \\
& | & acc_{\kappa}\langle t \rangle
\end{array}
$$

$$
\begin{array}{lll}
P, Q & ::= & \\
& & \mathbf{nil} \\
& | & \gamma.P \\
& | & (\nu\, s)P \\
& | & P \;|\; Q \\
& | & !P \\
& | & \ldots
\end{array}
$$

$$
\begin{array}{lll}
T & ::= & \\
& & \varepsilon \\
& | & v \\
& | & ?x \\
& | & \neg v \\
& | & T, T
\end{array}
$$

The semantics of $\mathcal{K}oS$ is defined by the relation

$$N \xrightarrow[\kappa]{\alpha} M$$

which states that N performs α with a cost κ and becomes M.

Local transitions (communications, node or link creations) have unitary QoS value, while the only non-trivial QoS values appear on the transitions that spawn processes or show the presence of links.

$\mathcal{K}oS$ Semantics[1]

(PREF) $$s :: \gamma.P \xrightarrow[\mathbf{1}]{\gamma@s} s :: P,\ \gamma \notin \{node_\kappa\langle t\rangle, con_\kappa\langle s\rangle, acc_\kappa\langle s\rangle\}$$

(CON) $$\frac{N \xrightarrow[\mathbf{1}]{s\ con_\kappa\langle t\rangle} N' \qquad M \xrightarrow[\mathbf{1}]{t\ acc_{\kappa'}\langle s\rangle} M'}{N \parallel M \xrightarrow[\mathbf{1}]{\tau} N' \parallel M' \parallel s \overset{\kappa}{\frown} t}\ \kappa \leq \kappa'$$

(COMM) $$\frac{N \xrightarrow[\mathbf{1}]{s\ (T)} N' \qquad M \xrightarrow[\mathbf{1}]{s\ \mathfrak{t}} M' \qquad \bowtie(T, \mathfrak{t}) = \sigma}{N \parallel M \xrightarrow[\mathbf{1}]{\tau} N'\sigma \parallel M'}$$

(LINK) $$s \overset{\kappa}{\frown} t \xrightarrow[\kappa]{s\ link\ t} \mathbf{0}$$

(NODE) $$s :: node_{\kappa}\langle t \rangle.P \xrightarrow[\mathbf{1}]{node\langle t \rangle} s :: P \parallel s \overset{\kappa}{\frown} t \parallel t :: \mathbf{0},\ s \neq t$$

(PAR) $$\frac{N \xrightarrow[\kappa]{\alpha} N'}{N \parallel M \xrightarrow[\kappa]{\alpha} N' \parallel M} \text{ if } \begin{cases} \mathrm{bn}(\alpha) \cap \mathrm{fn}(M) = \emptyset \quad \wedge \\ (\mathtt{addr}(N') \setminus \mathtt{addr}(N)) \cap \mathtt{addr}(M) = \emptyset \end{cases}$$

Rule (NODE) allows a process allocated at s to use a name t as the address of a new node and to create a new link from s to t exposing the QoS value κ. The side condition of (PAR) prevents that new nodes (and links) are created by using addresses of existing nodes.

$\mathcal{K}oS$ Semantics[3]

(LEVAL) $$s :: \varepsilon_\kappa[Q]@s.P \xrightarrow[\mathbf{1}]{\tau} s :: P \parallel s :: Q$$

(ROUTE) $$\frac{N \xrightarrow[\kappa']{r\ \varepsilon^s_\kappa\langle P\rangle @t} N' \qquad M \xrightarrow[\kappa'']{r\ link\ r'} M' \qquad \kappa' \star \kappa'' \leq \kappa}{N \parallel M \xrightarrow[\kappa' \star \kappa'']{r'\ \varepsilon^s_\kappa\langle P\rangle @t} N' \parallel M'} \quad t \neq r, r'$$

(LAND) $$\frac{N \xrightarrow[\kappa']{r\ \varepsilon^s_\kappa\langle P\rangle @t} N' \qquad M \xrightarrow[\kappa'']{r\ link\ t} M' \qquad \kappa' \star \kappa'' \leq \kappa}{N \parallel M \xrightarrow[\kappa' \star \kappa'']{\tau} N' \parallel M' \parallel t :: P}$$

Local spawning is always enabled while $\varepsilon_\kappa[P]@t$ from s is not always possible: the net must contain a path of links from s to t suitable wrt κ.

(ROUTE) states that P can traverse a link go an intermediate node r provided that costs are respected.

(LAND) describes the last hop: in this case, P is spawned at t, provided that the QoS value of the whole path that has been found is lower than κ.

Public links

Links in $\mathcal{K}o\mathcal{S}$ are public:

$$N \triangleq s :: \varepsilon_3[P]@t \ \| \ s \overset{1}{\frown} r \ \| \ r :: con_2\langle t\rangle.\varepsilon_2[Q]@t \ \| \ t :: acc_2\langle r\rangle,$$

- s and r are trying to spawn a process on t (but no path to t exists).
- r is aware that a link must be first created (and t agrees on that).

Initially, only (CON) con be applied:

$$N' \triangleq s :: \varepsilon_3[P]@t \ \| \ s \overset{1}{\frown} r \ \| \ r :: \varepsilon_2[Q]@t \ \| \ r \overset{2}{\frown} t \ \| \ t :: \mathbf{nil}.$$

$r \overset{2}{\frown} t$ provides now a path (costing 3) from s to t, hence using (PREF), (LINK), (ROUTE) and (LAND) we derive

$$N' \xrightarrow[3]{\tau} s :: \mathbf{nil} \ \| \ r :: \varepsilon_2[Q]@t \ \| \ t :: P.$$

Noteworthy, the migration of P prevents Q to be spawned because the link created by r has been used by P.

Private links

Private links can be traversed only by those processes having the appropriate "rights". Access rights are (particular) names.

$$N \triangleq s :: \varepsilon_{\{r,s\}}[P]@t \parallel s \overset{\{r\}}{\frown} s'$$

$$M \triangleq s :: \varepsilon_{\{r,s\}}[P]@t \parallel s \overset{\{r,u\}}{\frown} s'$$

P can traverse the link in N but not in M

Access rights c-semiring: $\mathcal{R} = \langle \wp_{\text{fin}}(\mathcal{S}) \cup \{\mathcal{S}\}, glb, \cup, \mathcal{S}, \emptyset \rangle$

$$X \leq Y \iff Y \subseteq X$$

A private link between the nodes s and t can be specified as

$$(\nu\, p)(s :: P \parallel s \overset{\{p\}}{\frown} t \parallel t :: Q)$$

Permanent and stable links

$\mathcal{K}oS$ links are vanishing but permanent links can be easily encoded:

$$s :: !con_{\kappa}\langle t\rangle \parallel t :: !acc_{\kappa'}\langle s\rangle$$

A slight variation are stable links, which are links existing until a given condition is satisfied.

$$Stable_s\ G\ t \ \triangleq\ !con_{\kappa}\langle t\rangle \ \mid\ \varepsilon[While\ G\ do\ acc_{\kappa}\langle s\rangle\ od\ \mathbf{nil}]@t$$

Conclusions

"Run, rabbit run
Dig that hole, forget the sun
And when at last the work is done
Don't sit down it's time to dig another one

(Breathe, Roger Waters)

Conclusions

We presented $\mathcal{K}oS$

- $\mathcal{K}oS$ aims at conveing the idea that QoS aspects are important for SOC applications
- $\mathcal{K}oS$ formally exploits c-semirings for representing QoS aspects
- c-semirings accounts for uniform handling of multicriteria QoS

Future work

- Further development of $\mathcal{K}oS$ theory e.g., observational semantics for $\mathcal{K}oS$ based on the idea of observing QoS values
- Equipping $\mathcal{K}oS$ with a type systems
  - having dependent types on links and their costs
  - types for access control to deal with QoS attributes
  - types for capturing the notion of contract
- Including $\mathcal{K}oS$ features in existing Klaim implementations
- In the paper we handled the QoS composition in overlay networks We intend to extend $\mathcal{K}oS$ with more general mechanisms for composing overlay networks than simple parallel composition via links

References

[BMR95] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Constraint solving over semiring. In *Proceedings of IJCAI95*, San Matco, 1995. CA: Morgan Kaufman.

[BMR97] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint satisfaction and optimization. *JACM*, 44(2):201–236, March 1997.

[DFM+03] Rocco De Nicola, Gianluigi Ferrari, Ugo Montanari, Rosario Pugliese, and Emilio Tuosto. A Formal Basis for Reasoning on Programmable QoS. In Nachum Dershowitz, editor, *International Symposium on Verification – Theory and Practice – Honoring Zohar Manna's 64th Birthday*, volume 2772 of *LNCS*, pages 436–479. Springer, 2003.

[HT05] Dan Hirsch and Emilio Tuosto. SHReQ: A Framework for Coordinating Application Level QoS. Submitted to SEFM, 2005.