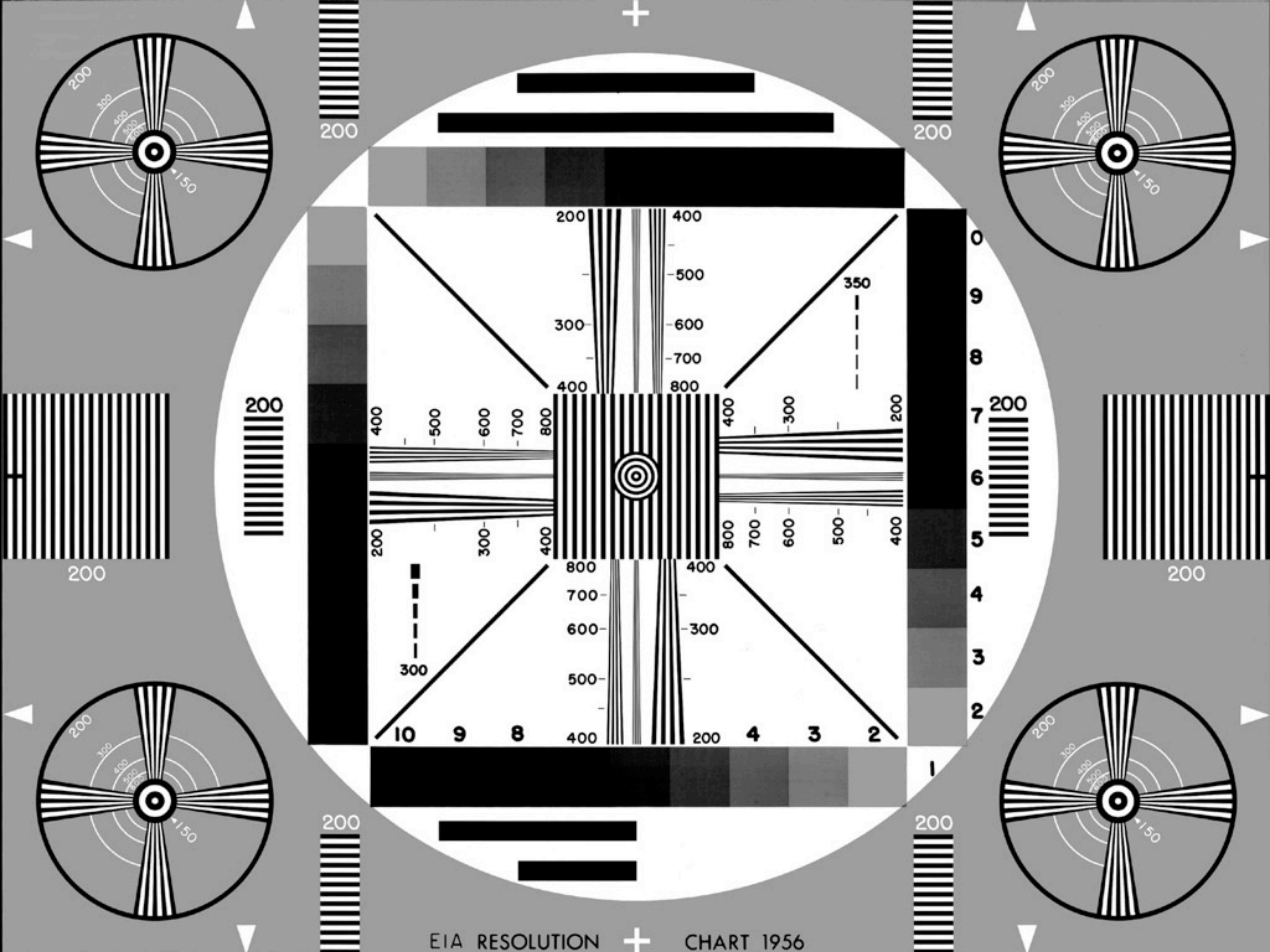

EIA RESOLUTION CHART 1956

This is not a pipe.

This is not a pipe.

This is not a pipe.

Theory of Diagnosis

Foundations
Extensions
Application

1.4. Bill of material :

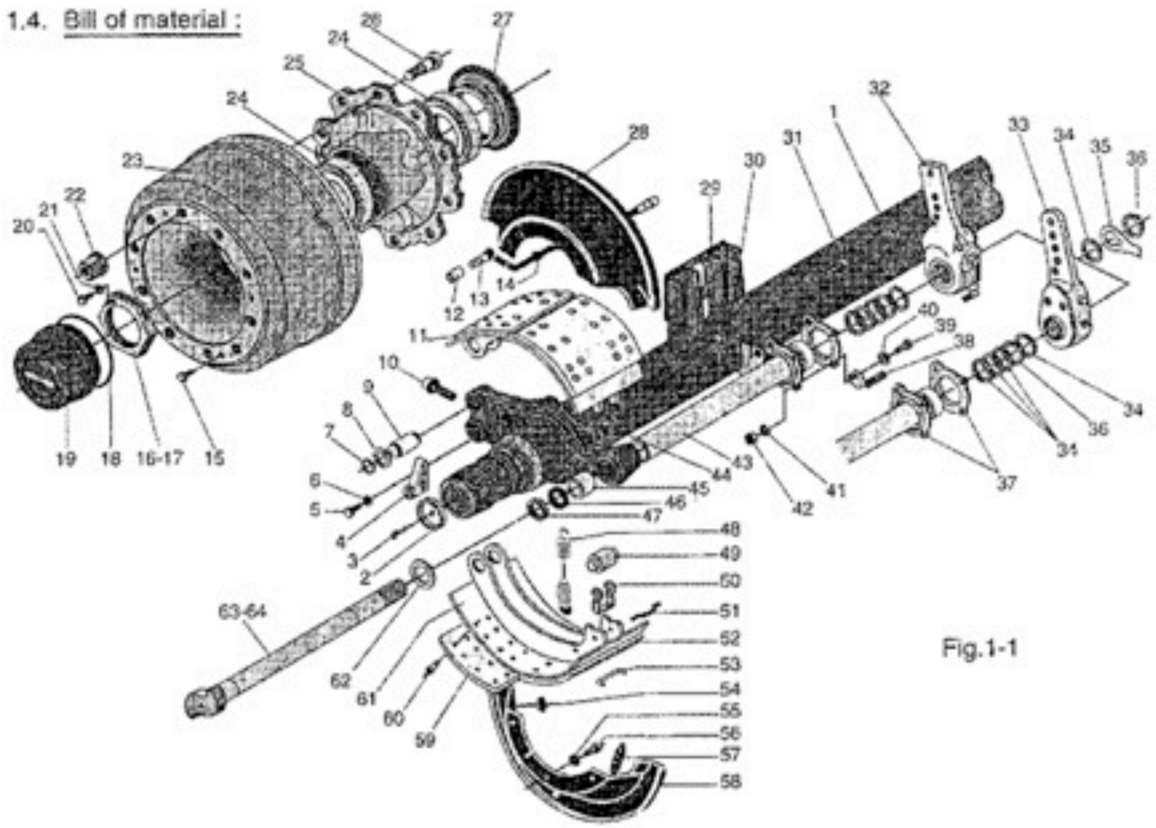


Fig.1-1

REP	NOMENCLATURE / BILL OF MATERIAL	REP	NOMENCLATURE / BILL OF MATERIAL
1	**Corps d'essieu assemblé** / Axle beam assembly	33	**Levier à réglage manuel** / Manual slack adjuster
2	**Bouchon** / Plug	34	**Rondelle** / Washer
3	**Goupille fendue** / Split pin	35	**Indicateur d'usure** / Wear indicator
4	**Support de capteur** / Sensor bracket	36	**Circlip** / Circlip
5	**Vis** / Bolt	37	**Cage de palier** / Camshaft bracket
6	**Rondelle** / Washer	38	**Patte de point fixe** / Fixed point assembly
7	**Circlip** / Circlip	39	**Vis** / Bolt
8	**Rondelle** / Washer	40	**Rondelle** / Washer
9	**Axe de machoire** / Anchor pin	41	**Rondelle** / Washer
10	**Vis de sécurité** / Bolt - security	42	**Ecrou** / Nut
11	**Machoire garnie** / Brake shoe assembly	43	**Fourreau d'arbre à came** / Camshaft bearing tube
12	**Bague de capteur** / Retaining ring	44	**Graisseur** / Nipple
13	**Capteur ABS** / ABS sensor	45	**Bague roulée** / Bushing
14	**Passe-fil** / Grommet	46	**Joint torique** / 'O' ring
15	**Vis de tambour** / Bolt	47	**Rondelle plastique** / Plastic washer
16	**Ecrou de fusée (pas à gauche)** / Axle nut (left hand thread)	48	**Ressort de rappel** / Return spring
17	**Ecrou de fusée (pas à droite)** / Axle nut (right hand thread)	49	**Galet** / Roller
18	**Joint torique** / 'O' ring	50	**Support de galet** / Retainer
19	**Chapeau de moyeu** / Hub cap	51	**Axe de ressort** / Pin
20	**Vis frein** / Locking bolt	52	**Garniture coté came** / Lining - cam
21	**Rondelle** / Washer	53	**Fixation de galet** / Pin roller retainer
22	**Ecrou de roue** / Wheel nut	54	**Bouchon plastique** / Plastic plug
23	**Tambour** / Drum	55	**Rondelle** / Washer
24	**Roulement** / Bearing	56	**Vis** / Bolt
25	**Moyeu** / Hub	57	**Bouchon caoutchouc** / Plug rubber
26	**Goujon de fixation de roue** / Wheel bolt	58	**Cache-tambour** / Dust cover
27	**Joint de moyeu** / Hub seal	59	**Garniture coté axe** / Lining - anchor
28	**Cache-tambour** / Dust cover	60	**Rivet** / Rivet
29	**Ecrou** / Bolt	61	**Machoire nue** / Brake shoe
30	**Rondelle** / Washer	62	**Rondelle** / Washer
31	**Vis** / Bolt	63	**Arbre à came** / Camshaft
32	**Levier à réglage automatique** / Automatic slack adjuster	64	**Arbre à came** / Camshaft

1.4. Bill of material :

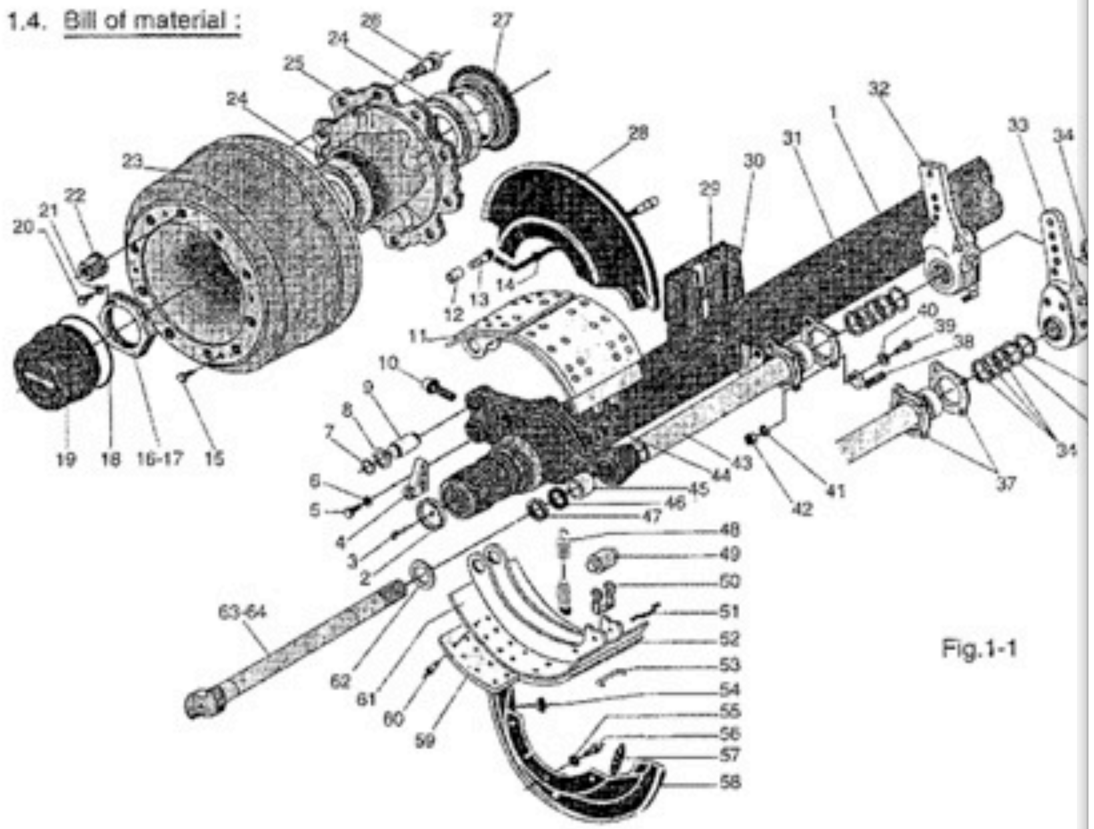


Fig.1-1

REP	NOMENCLATURE / BILL OF MATERIAL	REP	NOMENCLATURE / BILL OF MATERIAL
1	**Corps d'essieu assemblé** / Axle beam assembly	33	**Levier à réglage manuel** / Manual slack adju
2	**Bouchon** / Plug	34	**Rondelle** / Washer
3	**Goupille fendue** / Split pin	35	**Indicateur d'usure** / Wear indicator
4	**Support de capteur** / Sensor bracket	36	**Circlip** / Circlip
5	**Vis** / Bolt	37	**Cage de palier** / Camshaft bracket
6	**Rondelle** / Washer	38	**Patte de point fixe** / Fixed point assembly
7	**Circlip** / Circlip	39	**Vis** / Bolt
8	**Rondelle** / Washer	40	**Rondelle** / Washer
9	**Axe de machoire** / Anchor pin	41	**Rondelle** / Washer
10	**Vis de sécurité** / Bolt - security	42	**Ecrou** / Nut
11	**Machoire garnie** / Brake shoe assembly	43	**Fourreau d'arbre à came** / Camshaft bearing
12	**Bague de capteur** / Retaining ring	44	**Graisseur** / Nipple
13	**Capteur ABS** / ABS sensor	45	**Bague roulée** / Bushing
14	**Passe-fil** / Grommet	46	**Joint torique** / 'O' ring
15	**Vis de tambour** / Bolt	47	**Rondelle plastique** / Plastic washer
16	**Ecrou de fusée (pas à gauche)** / Axle nut (left hand thread)	48	**Ressort de rappel** / Return spring
17	**Ecrou de fusée (pas à droite)** / Axle nut (right hand thread)	49	**Galet** / Roller
18	**Joint torique** / 'O' ring	50	**Support de galet** / Retainer
19	**Chapeau de moyeu** / Hub cap	51	**Axe de ressort** / Pin
20	**Vis frein** / Locking bolt	52	**Garniture coté came** / Lining - cam
21	**Rondelle** / Washer	53	**Fixation de galet** / Pin roller retainer
22	**Ecrou de roue** / Wheel nut	54	**Bouchon plastique** / Plastic plug
23	**Tambour** / Drum	55	**Rondelle** / Washer
24	**Roulement** / Bearing	56	**Vis** / Bolt
25	**Moyeu** / Hub	57	**Bouchon caoutchouc** / Plug rubber
26	**Goujon de fixation de roue** / Wheel bolt	58	**Cache-tambour** / Dust cover
27	**Joint de moyeu** / Hub seal	59	**Garniture coté axe** / Lining - anchor
28	**Cache-tambour** / Dust cover	60	**Rivet** / Rivet
29	**Ecrou** / Bolt	61	**Machoire nue** / Brake shoe
30	**Rondelle** / Washer	62	**Rondelle** / Washer
31	**Vis** / Bolt	63	**Arbre à came** / Camshaft
32	**Levier à réglage automatique** / Automatic slack adjuster	64	**Arbre à came** / Camshaft

QUOTE

Company Name

625 E Beale
Kingman, AZ 86401
Ph 928-753-6945 Fax 928-718-3238
craig@mesc.org

Quote Number 1001
Quote Date March 7, 2008
Project XYZ School District Wobulator

TO [Customer Representative]
[Customer Name]
[Street Address]
[City, ST ZIP Code]
[Phone]

NOTES: [Place notes about the project here, e.g., "customer is responsible for removal of old product"]

SALESPERSON	PAYMENT TERMS	EXPECTED DELIVERY	QUOTE VALID:
Craig McKee	As per MESC contract listed below	30 days ARO	30 Days from date above

QTY	PART #	DESCRIPTION	UNIT PRICE	DISC	MESC PRICE	LINE TOTAL
10.00	123WS	Wobulator Shaft	$100.00	10%	$90.00	$900.00
1.00	123MWS	Mega Wobulator Shaft	$150.00	10%	$135.00	$135.00

MISCELLANEOUS	
SUBTOTAL	$ 1,035.00
SALES TAX%	0.08
TOTAL	**$ 1,117.80**

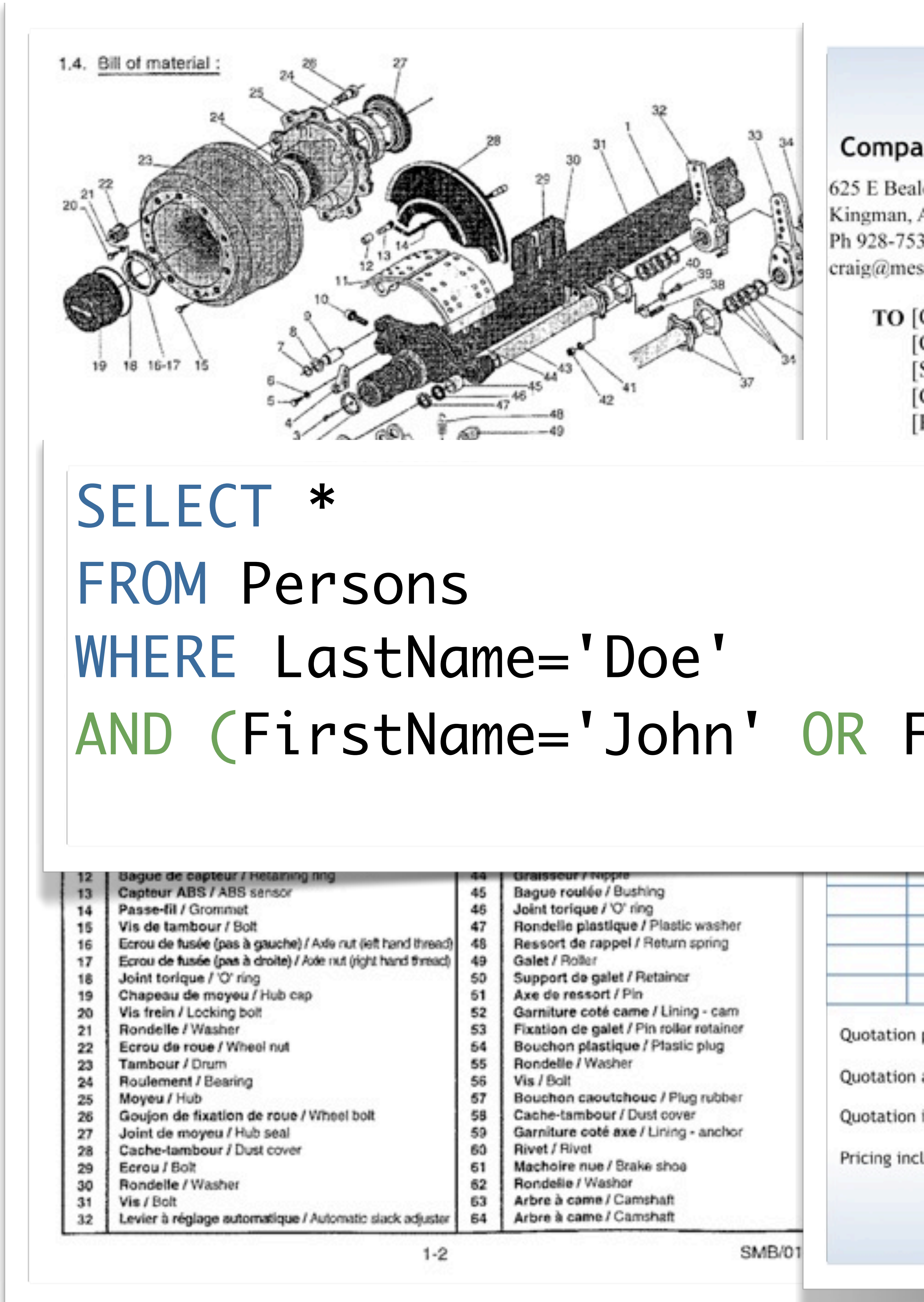
Quotation prepared by: [Sales Representative]

Quotation accepted by: [Member Representative]

Quotation is subject to the conditions of MESC contract: 08X-XXX-1234

Pricing includes MESC Admin Fee

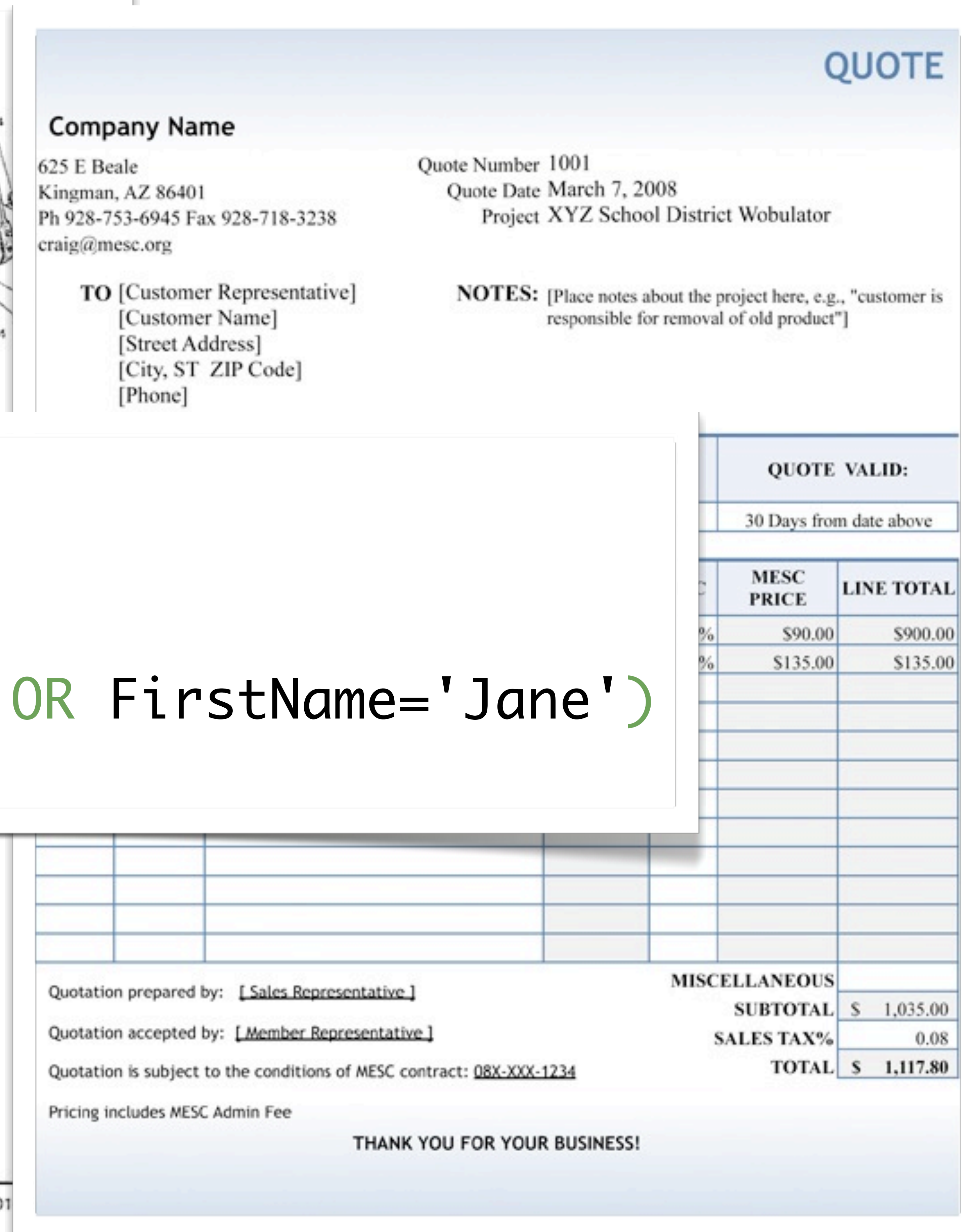
THANK YOU FOR YOUR BUSINESS!



1.4. Bill of material :

12	Bague de capteur / Retaining ring	44	Graisseur / Nipple
13	Capteur ABS / ABS sensor	45	Bague roulée / Bushing
14	Passe-fil / Grommet	46	Joint torique / 'O' ring
15	Vis de tambour / Bolt	47	Rondelle plastique / Plastic washer
16	Ecrou de fusée (pas à gauche) / Axle nut (left hand thread)	48	Ressort de rappel / Return spring
17	Ecrou de fusée (pas à droite) / Axle nut (right hand thread)	49	Galet / Roller
18	Joint torique / 'O' ring	50	Support de galet / Retainer
19	Chapeau de moyeu / Hub cap	51	Axe de ressort / Pin
20	Vis frein / Locking bolt	52	Garniture coté came / Lining - cam
21	Rondelle / Washer	53	Fixation de galet / Pin roller retainer
22	Ecrou de roue / Wheel nut	54	Bouchon plastique / Plastic plug
23	Tambour / Drum	55	Rondelle / Washer
24	Roulement / Bearing	56	Vis / Bolt
25	Moyeu / Hub	57	Bouchon caoutchouc / Plug rubber
26	Goujon de fixation de roue / Wheel bolt	58	Cache-tambour / Dust cover
27	Joint de moyeu / Hub seal	59	Garniture coté axe / Lining - anchor
28	Cache-tambour / Dust cover	60	Rivet / Rivet
29	Ecrou / Bolt	61	Machoire nue / Brake shoe
30	Rondelle / Washer	62	Rondelle / Washer
31	Vis / Bolt	63	Arbre à came / Camshaft
32	Levier à réglage automatique / Automatic slack adjuster	64	Arbre à came / Camshaft

1-2 SMB/01



QUOTE

Company Name

625 E Beale
Kingman, AZ 86401
Ph 928-753-6945 Fax 928-718-3238
craig@mesc.org

Quote Number 1001
Quote Date March 7, 2008
Project XYZ School District Wobulator

TO [Customer Representative]
[Customer Name]
[Street Address]
[City, ST ZIP Code]
[Phone]

NOTES: [Place notes about the project here, e.g., "customer is responsible for removal of old product"]

QUOTE VALID: 30 Days from date above

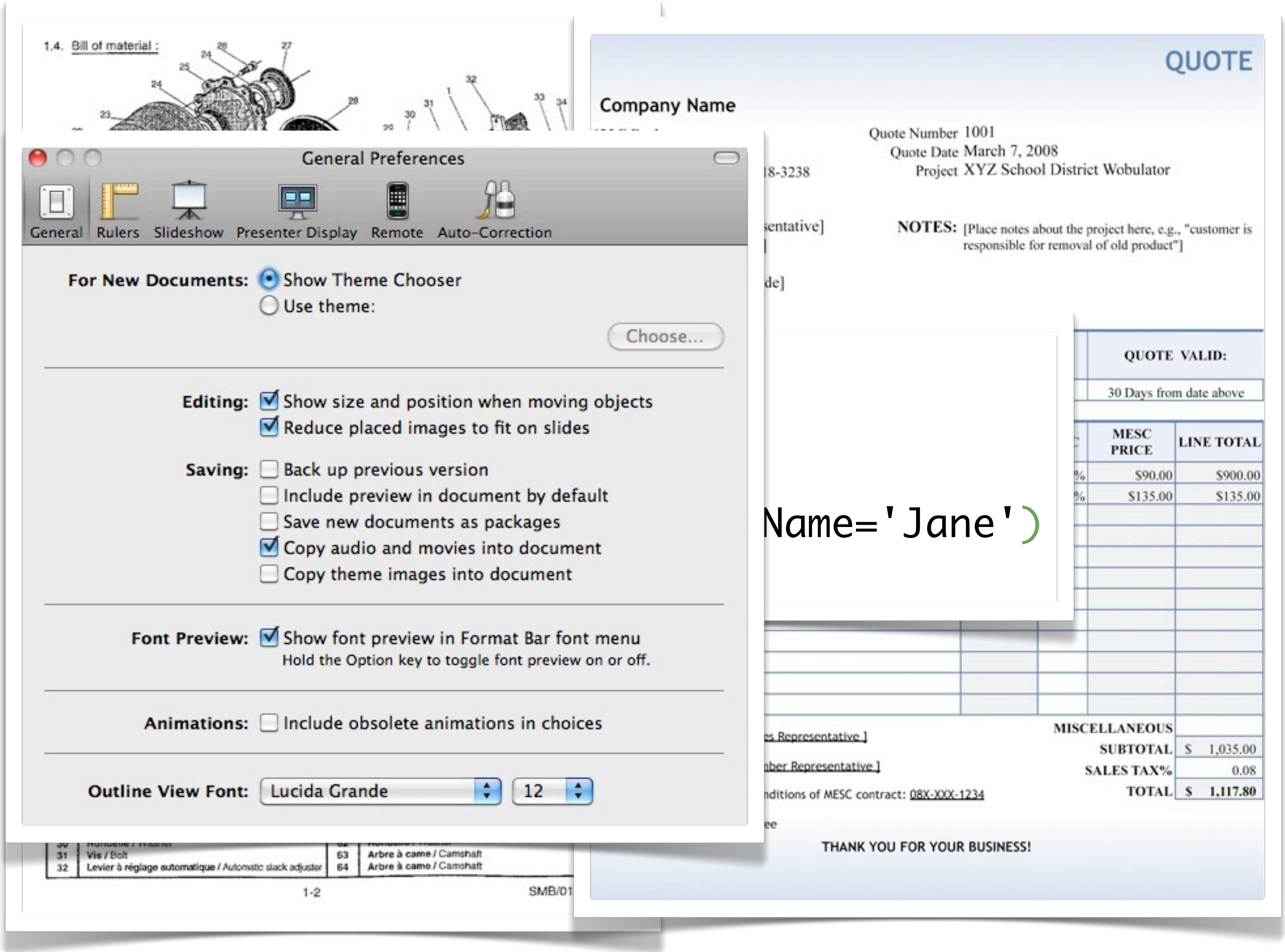
MESC PRICE	LINE TOTAL
$90.00	$900.00
$135.00	$135.00

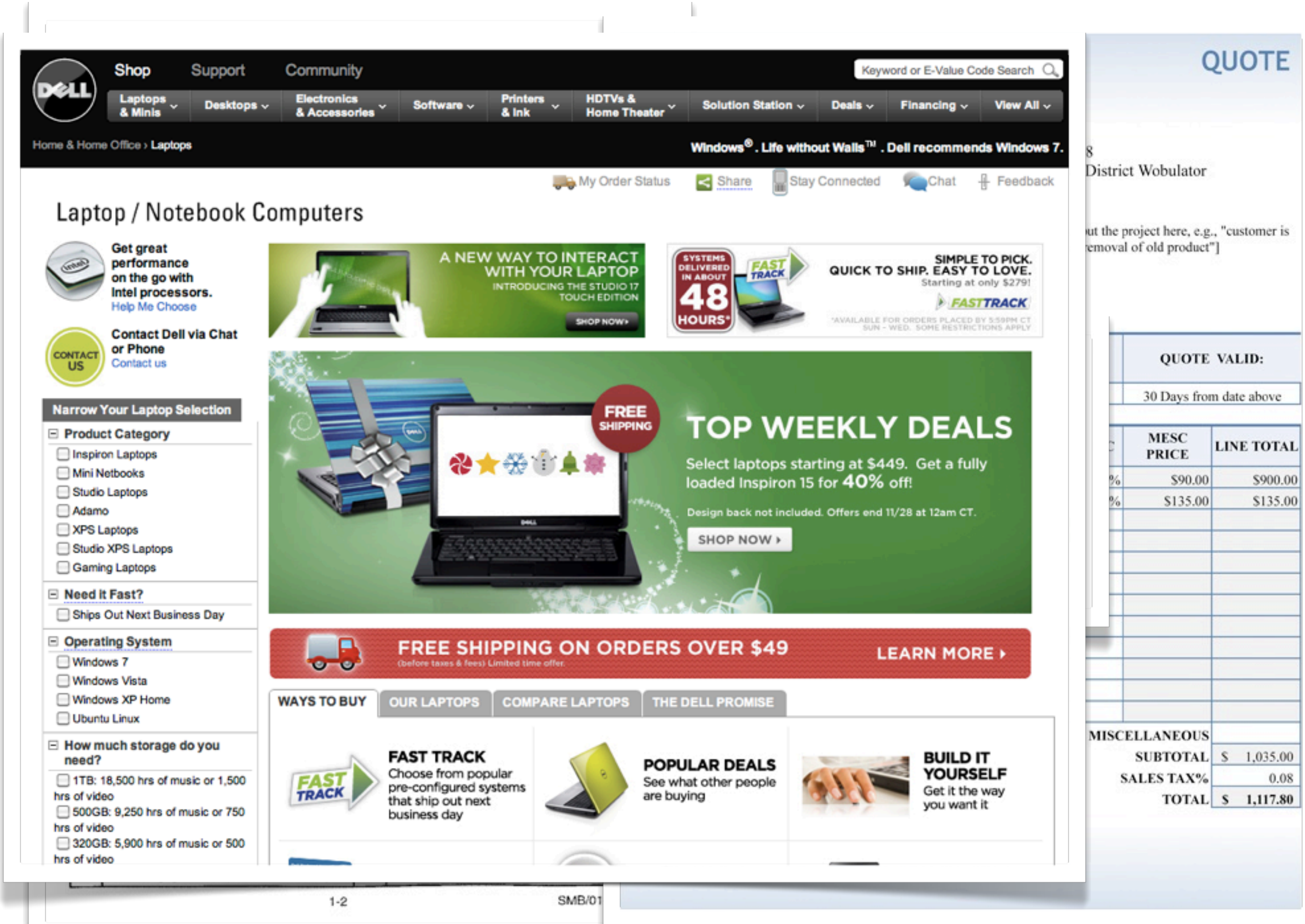
Quotation prepared by: [Sales Representative]
Quotation accepted by: [Member Representative]
Quotation is subject to the conditions of MESC contract: 08X-XXX-1234
Pricing includes MESC Admin Fee

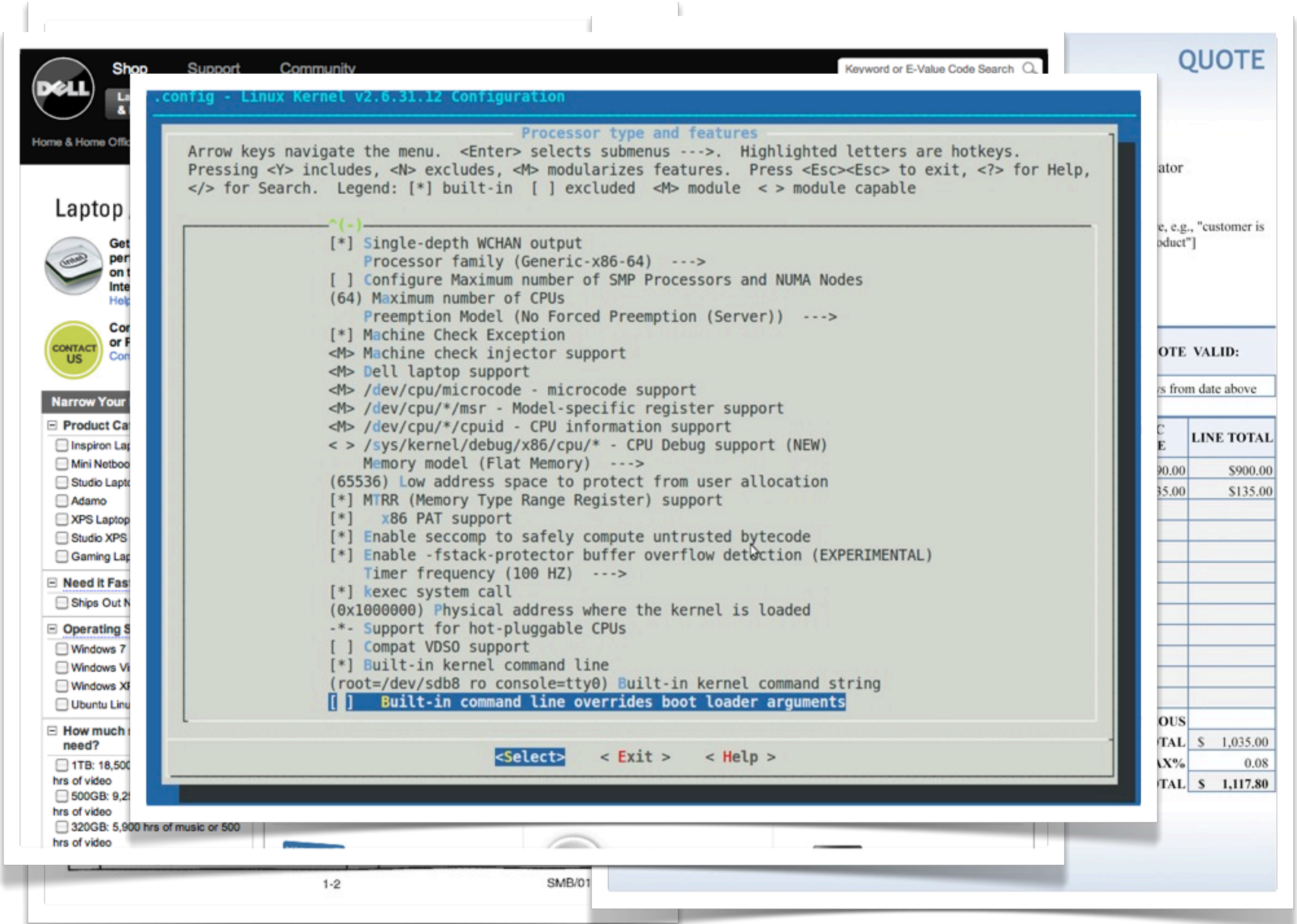
MISCELLANEOUS	
SUBTOTAL	$ 1,035.00
SALES TAX%	0.08
TOTAL	$ 1,117.80

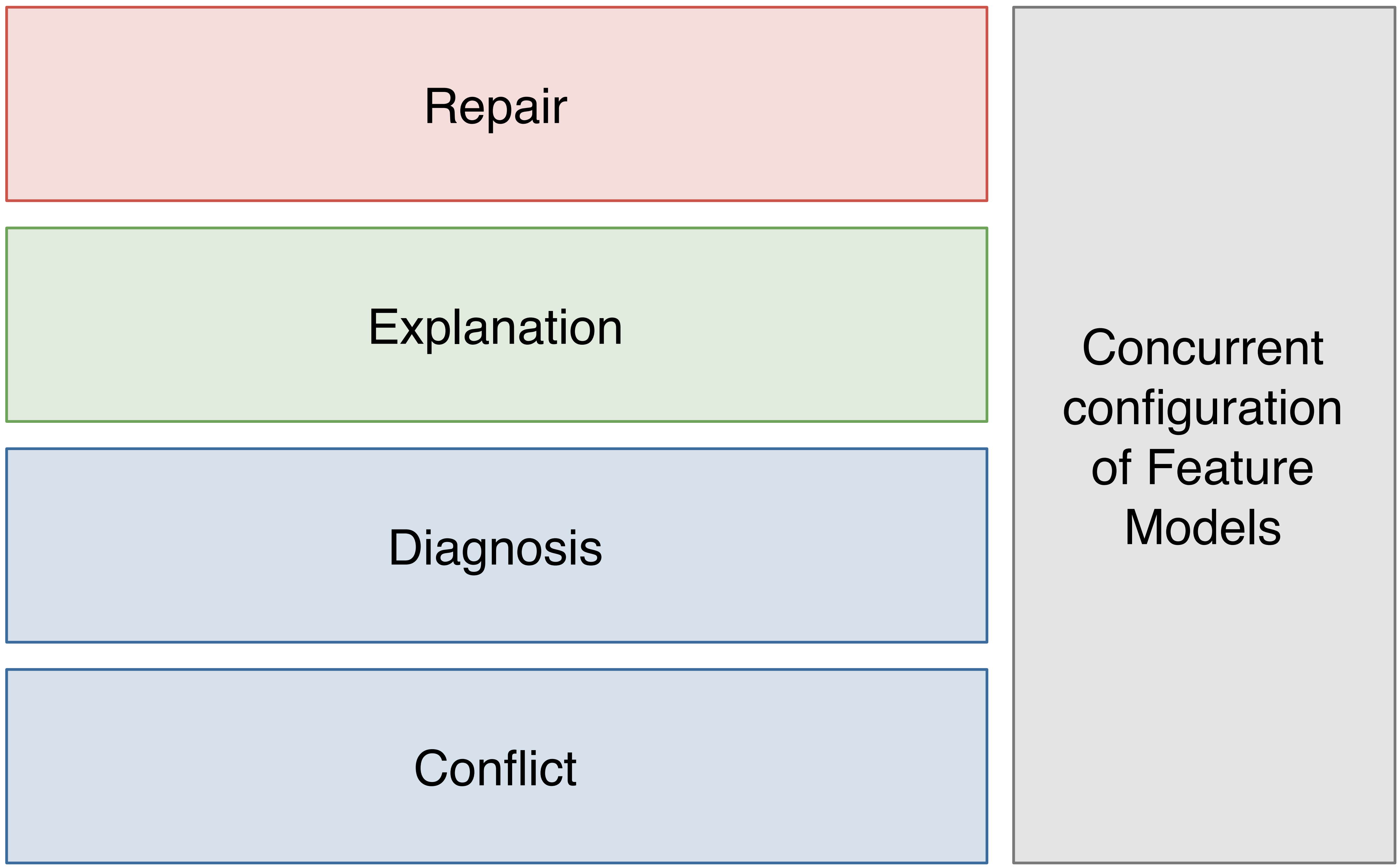
THANK YOU FOR YOUR BUSINESS!

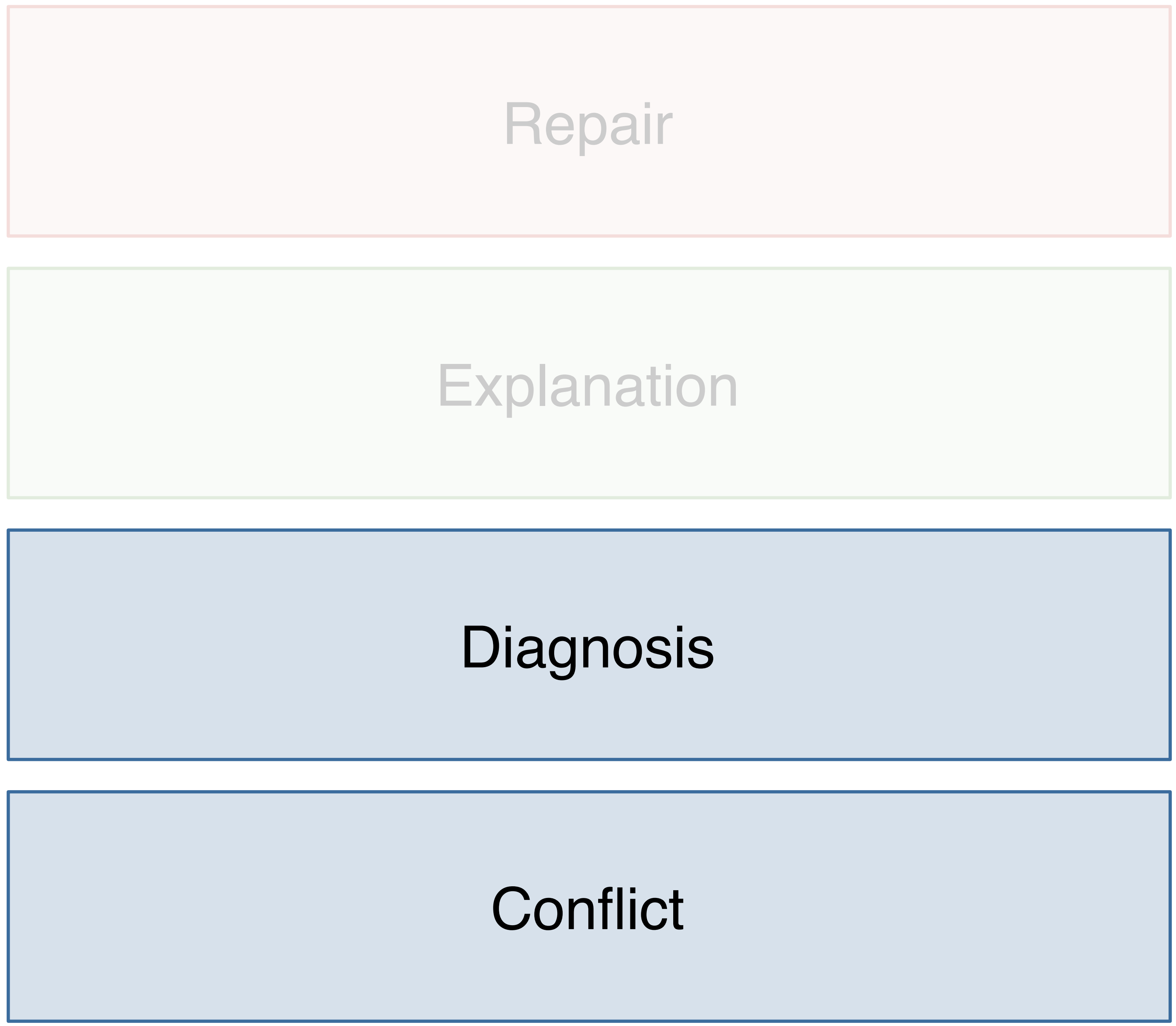
```sql
SELECT *
FROM Persons
WHERE LastName='Doe'
AND (FirstName='John' OR FirstName='Jane')
```


1.4. Bill of material :
QUOTE
Company Name
Quote Number 1001
Quote Date March 7, 2008
Project XYZ School District Wobulator
NOTES: [Place notes about the project here, e.g., "customer is responsible for removal of old product"]
General Preferences
General
Rulers
Slideshow
Presenter Display
Remote
Auto-Correction
For New Documents:
Show Theme Chooser
Use theme:
Choose...
Editing:
Show size and position when moving objects
Reduce placed images to fit on slides
Saving:
Back up previous version
Include preview in document by default
Save new documents as packages
Copy audio and movies into document
Copy theme images into document
Font Preview:
Show font preview in Format Bar font menu
Hold the Option key to toggle font preview on or off.
Animations:
Include obsolete animations in choices
Outline View Font:
Lucida Grande
12
Name='Jane')
QUOTE VALID:
30 Days from date above
MESC PRICE
LINE TOTAL
$90.00
$900.00
$135.00
$135.00
MISCELLANEOUS
SUBTOTAL
$ 1,035.00
SALES TAX%
0.08
TOTAL
$ 1,117.80
THANK YOU FOR YOUR BUSINESS!
31 Vis / Bolt
32 Levier à réglage automatique / Automatic slack adjuster
63 Arbre à came / Camshaft
64 Arbre à came / Camshaft
1-2
SMB/01


Shop
Support
Community
Keyword or E-Value Code Search
Laptops & Minis
Desktops
Electronics & Accessories
Software
Printers & Ink
HDTVs & Home Theater
Solution Station
Deals
Financing
View All
Home & Home Office › Laptops
Windows®. Life without Walls™. Dell recommends Windows 7.
My Order Status
Share
Stay Connected
Chat
Feedback
Laptop / Notebook Computers
Get great performance on the go with Intel processors.
Help Me Choose
Contact Dell via Chat or Phone
Contact us
CONTACT US
A NEW WAY TO INTERACT WITH YOUR LAPTOP
INTRODUCING THE STUDIO 17 TOUCH EDITION
SHOP NOW›
SYSTEMS DELIVERED IN ABOUT 48 HOURS*
FAST TRACK
SIMPLE TO PICK. QUICK TO SHIP. EASY TO LOVE.
Starting at only $279!
FASTTRACK
Narrow Your Laptop Selection
Product Category
Inspiron Laptops
Mini Netbooks
Studio Laptops
Adamo
XPS Laptops
Studio XPS Laptops
Gaming Laptops
Need it Fast?
Ships Out Next Business Day
Operating System
Windows 7
Windows Vista
Windows XP Home
Ubuntu Linux
How much storage do you need?
1TB: 18,500 hrs of music or 1,500 hrs of video
500GB: 9,250 hrs of music or 750 hrs of video
320GB: 5,900 hrs of music or 500 hrs of video
FREE SHIPPING
TOP WEEKLY DEALS
Select laptops starting at $449. Get a fully loaded Inspiron 15 for 40% off!
Design back not included. Offers end 11/28 at 12am CT.
SHOP NOW ›
FREE SHIPPING ON ORDERS OVER $49
LEARN MORE ›
WAYS TO BUY
OUR LAPTOPS
COMPARE LAPTOPS
THE DELL PROMISE
FAST TRACK
Choose from popular pre-configured systems that ship out next business day
POPULAR DEALS
See what other people are buying
BUILD IT YOURSELF
Get it the way you want it
1-2
SMB/01
QUOTE
District Wobulator
QUOTE VALID:
30 Days from date above
MESC PRICE
LINE TOTAL
$90.00
$900.00
$135.00
$135.00
MISCELLANEOUS
SUBTOTAL
$ 1,035.00
SALES TAX%
0.08
TOTAL
$ 1,117.80


.config - Linux Kernel v2.6.31.12 Configuration
Processor type and features
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
^(-)
[*] Single-depth WCHAN output
Processor family (Generic-x86-64) --->
[ ] Configure Maximum number of SMP Processors and NUMA Nodes
(64) Maximum number of CPUs
Preemption Model (No Forced Preemption (Server)) --->
[*] Machine Check Exception
<M> Machine check injector support
<M> Dell laptop support
<M> /dev/cpu/microcode - microcode support
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
< > /sys/kernel/debug/x86/cpu/* - CPU Debug support (NEW)
Memory model (Flat Memory) --->
(65536) Low address space to protect from user allocation
[*] MTRR (Memory Type Range Register) support
[*] x86 PAT support
[*] Enable seccomp to safely compute untrusted bytecode
[*] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
Timer frequency (100 HZ) --->
[*] kexec system call
(0x1000000) Physical address where the kernel is loaded
-*- Support for hot-pluggable CPUs
[ ] Compat VDSO support
[*] Built-in kernel command line
(root=/dev/sdb8 ro console=tty0) Built-in kernel command string
[ ] Built-in command line overrides boot loader arguments
<Select> < Exit > < Help >
Shop
Support
Community
Keyword or E-Value Code Search
Home & Home Offic
Laptop
CONTACT US
Narrow Your
Product Ca
Inspiron Lap
Mini Netboo
Studio Lapto
Adamo
XPS Laptop
Studio XPS
Gaming Lap
Need It Fas
Ships Out N
Operating S
Windows 7
Windows Vi
Windows XI
Ubuntu Linu
How much need?
1TB: 18,500 hrs of video
500GB: 9,2 hrs of video
320GB: 5,900 hrs of music or 500 hrs of video
1-2
SMB/01
QUOTE
LINE TOTAL
$900.00
$135.00
$ 1,035.00
0.08
$ 1,117.80


Repair
Explanation
Diagnosis
Conflict
Concurrent configuration of Feature Models


Repair
Explanation
Diagnosis
Conflict

Reiter's Theory of Diagnosis [Reiter'87]

Definition *System* [Reiter'87,Def 2.1]

A *system* is a pair (SD,COMPONENTS) where:

1. SD, the system description, is a set of first-order sentences;
2. COMPONENTS, the system components, is a finite set of constants.

Reiter's Theory of Diagnosis [Reiter'87]

Definition *System* [Reiter'87,Def 2.1]

A *system* is a pair (SD,COMPONENTS) where:

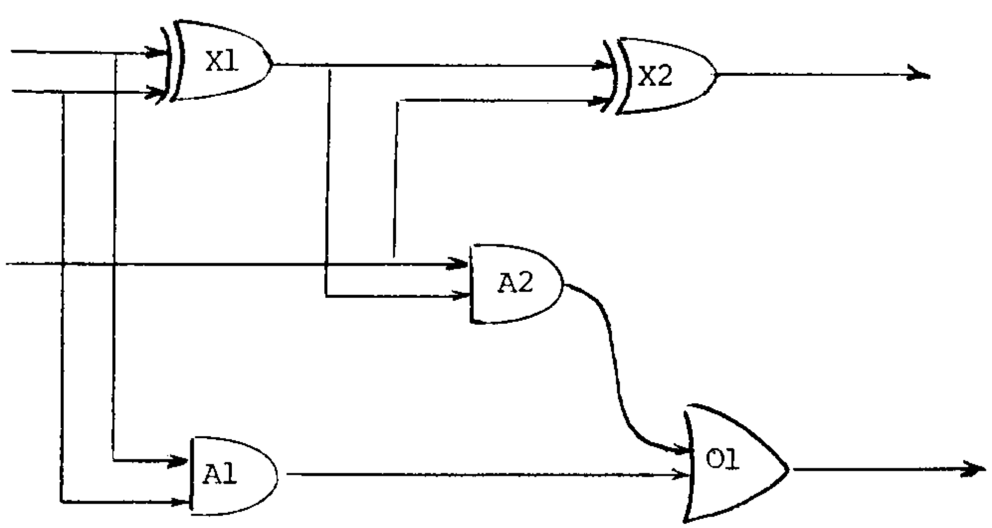
1. SD, the system description, is a set of first-order sentences;
2. COMPONENTS, the system components, is a finite set of constants.

Definition *Abnormality* [Reiter'87]

The unary predicate $\mathrm{AB}(\cdot)$ denotes the abnormality of a component in a given system description.

Reiter's Theory of Diagnosis [Reiter'87]

Example *Binary full adder* [Reiter'87]



SD

$$\text{ANDG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{and}(\text{in1}(x), \text{in2}(x)) ,$$
$$\text{XORG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{xor}(\text{in1}(x), \text{in2}(x)) ,$$
$$\text{ORG}(x) \wedge \neg\text{AB}(x) \supset \text{out}(x) = \text{or}(\text{in1}(x), \text{in2}(x)) ,$$
$$\text{ANDG}(A_1), \quad \text{ANDG}(A_2) ,$$
$$\text{XORG}(X_1) , \quad \text{XORG}(X_2) \quad \text{ORG}(O_1) ,$$

$$\text{out}(X_1) = \text{in2}(A_2) ,$$
$$\text{out}(X_1) = \text{in1}(X_2) ,$$
$$\text{out}(A_2) = \text{in1}(O_1) ,$$
$$\text{in1}(A_2) = \text{in2}(X_2) ,$$
$$\text{in1}(X_1) = \text{in1}(A_1) ,$$
$$\text{in2}(X_1) = \text{in2}(A_1) ,$$
$$\text{out}(A_1) = \text{in2}(O_1) .$$

COMPONENTS

$$\{A_1, A_2, X_1, X_2, O_1\}$$

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Observation* [Reiter'87,Def 2.3]

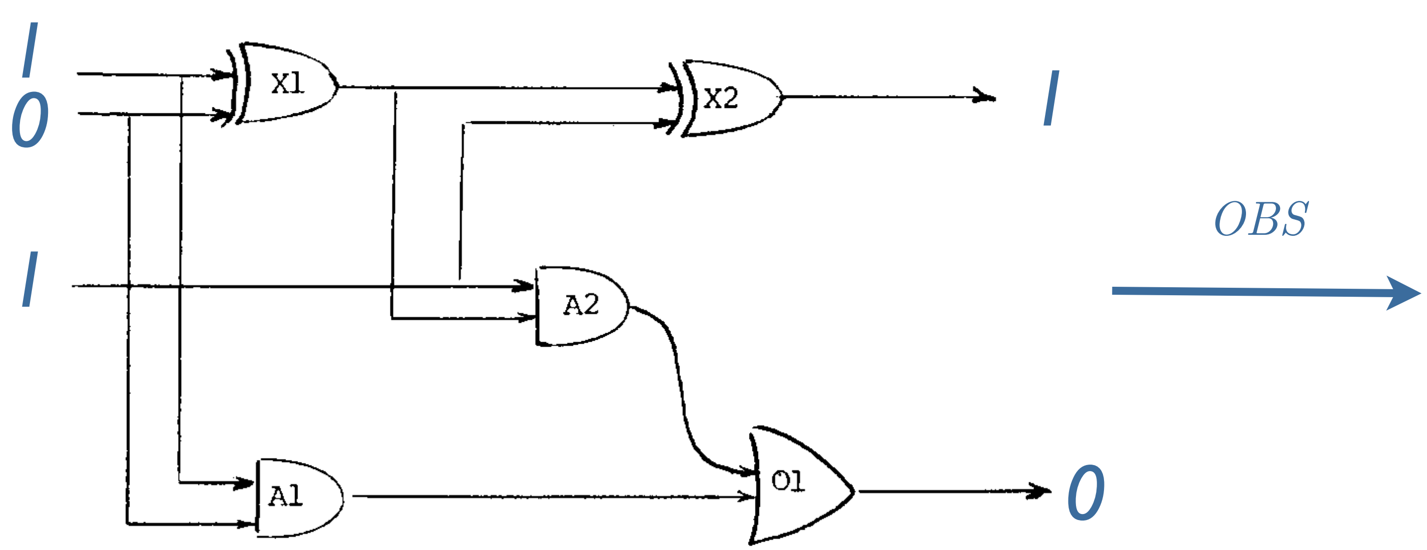
An *observation* of a system is a finite set of first-order sentences. The triple (SD, COMPONENTS,OBS) denotes the system (SD, COMPONENTS) with observation OBS.

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Observation* [Reiter'87,Def 2.3]

An *observation* of a system is a finite set of first-order sentences. The triple (SD, COMPONENTS,OBS) denotes the system (SD, COMPONENTS) with observation OBS.

Example *Binary full adder* [Reiter'87]



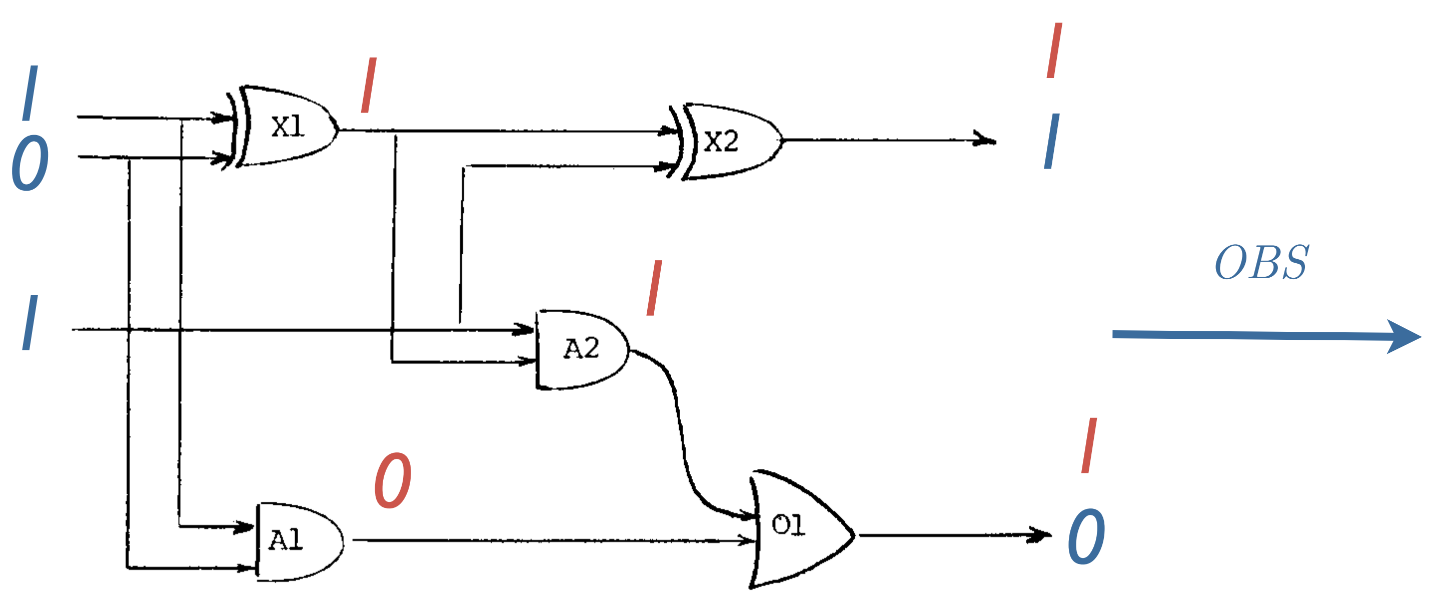
$$\mathrm{in1}(X_1) = 1\,,$$
$$\mathrm{in2}(X_1) = 0\,,$$
$$\mathrm{in1}(A_2) = 1\,,$$
$$\mathrm{out}(X_2) = 1\,,$$
$$\mathrm{out}(O_1) = 0\,.$$

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Observation* [Reiter'87,Def 2.3]

An *observation* of a system is a finite set of first-order sentences. The triple (SD, COMPONENTS,OBS) denotes the system (SD, COMPONENTS) with observation OBS.

Example *Binary full adder* [Reiter'87]



$\text{in1}(X_1) = 1$,

$\text{in2}(X_1) = 0$,

$\text{in1}(A_2) = 1$,

$\text{out}(X_2) = 1$,

$\text{out}(O_1) = 0$.

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Diagnosis* [Reiter'87,Def 2.3]

A *diagnosis* for (SD, COMPONENTS,OBS) is a minimal set $\Delta \subset$COMPONENTS such that:

$SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPONENTS - \Delta\}$

is consistent.

Reiter's Theory of Diagnosis [Reiter'87]

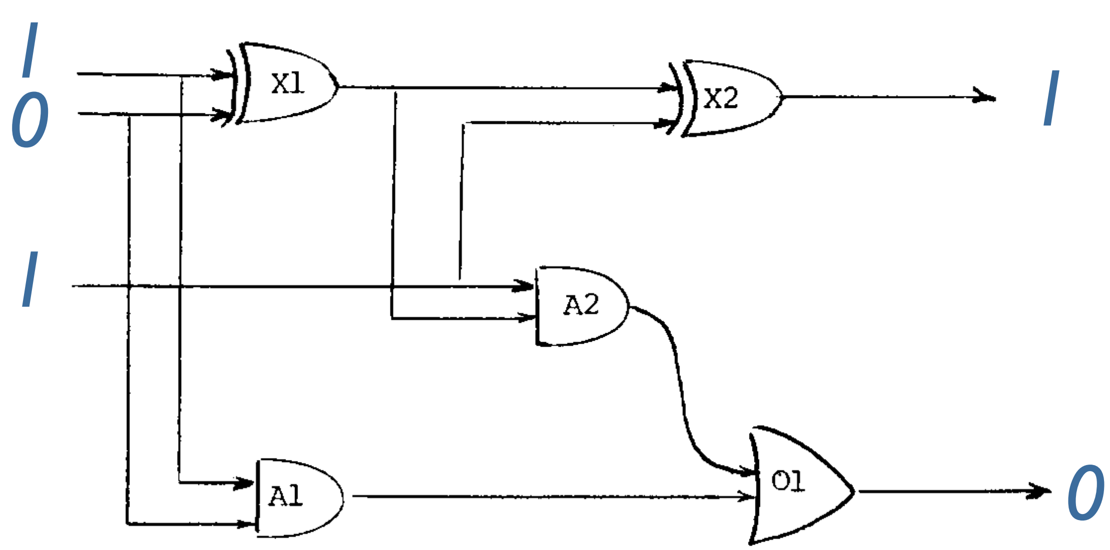
Definition *Diagnosis* [Reiter'87,Def 2.3]

A *diagnosis* for (SD, COMPONENTS,OBS) is a minimal set $\Delta \subset$COMPONENTS such that:

$$SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPONENTS - \Delta\}$$

is consistent.

Example *Binary full adder* [Reiter'87]



$\xrightarrow{\Delta}$

$\{X_1\}$,
$\{X_2, O_1\}$,
$\{X_2\ A_2\}$,

Reiter's Theory of Diagnosis [Reiter'87]

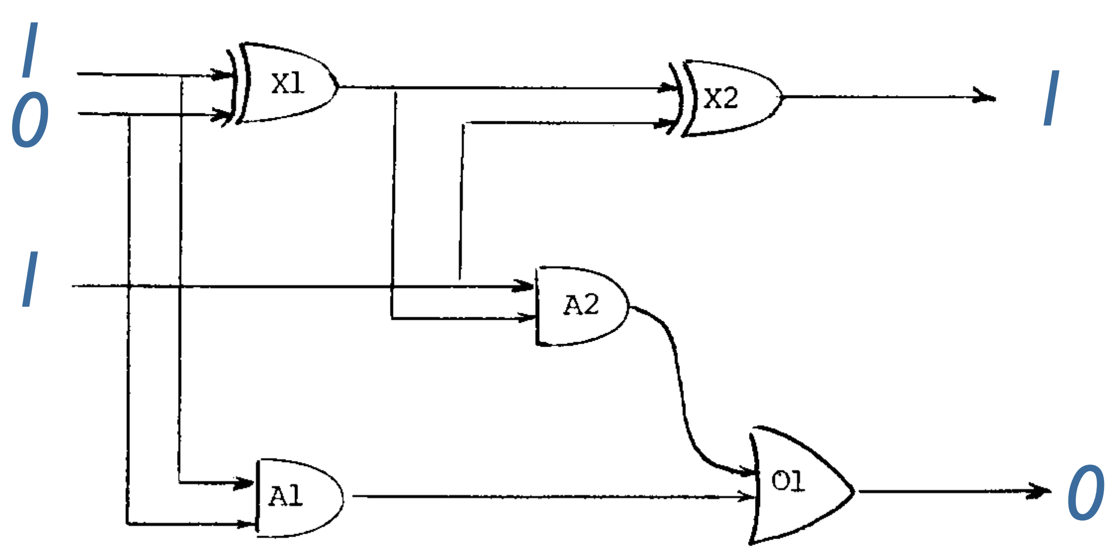
Definition *Diagnosis* [Reiter'87,Def 2.3]

A *diagnosis* for (SD, COMPONENTS,OBS) is a minimal set $\Delta \subset$ COMPONENTS such that:

$$SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPONENTS - \Delta\}$$

is consistent.

Example *Binary full adder* [Reiter'87]



Δ

smallest sets

$\{X_1\}$,
$\{X_2, O_1\}$,
$\{X_2\ A_2\}$,

Reiter's Theory of Diagnosis [Reiter'87]

Proposition *Diagnosis existence* [Reiter'87,Prop 3.1]

A diagnosis *exists* for (SD, COMPONENTS,OBS) iff SD∪OBS is consistent.

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Conflict set* [Reiter'87,Def 4.1]

A *conflict set* for (SD, COMPONENTS,OBS) is a set $\{c_1,\ldots,c_k\} \subseteq$ COMPONENTS such that:

$$SD \cup OBS \cup \{\neg AB(c_1),\ldots, \neg AB(c_k)\}$$

is **in**consistent.

Reiter's Theory of Diagnosis [Reiter'87]

Definition *Conflict set* [Reiter'87,Def 4.1]

A *conflict set* for (SD, COMPONENTS,OBS) is a set $\{c_1,\ldots,c_k\} \subseteq$ COMPONENTS such that:

$$SD \cup OBS \cup \{\neg AB(c_1),\ldots, \neg AB(c_k)\}$$

is **in**consistent.

Definition *Hitting set* [Reiter'87,Def 4.3]

Suppose C is a collection of sets. A *hitting set* for C is a set $H \subseteq_{S \in C} S$ such that $H \cap S \neq \{\}$ for each $S \in C$. A hitting set for C is *minimal* iff no proper subset of it is a hitting set for C.

Reiter's Theory of Diagnosis [Reiter'87]

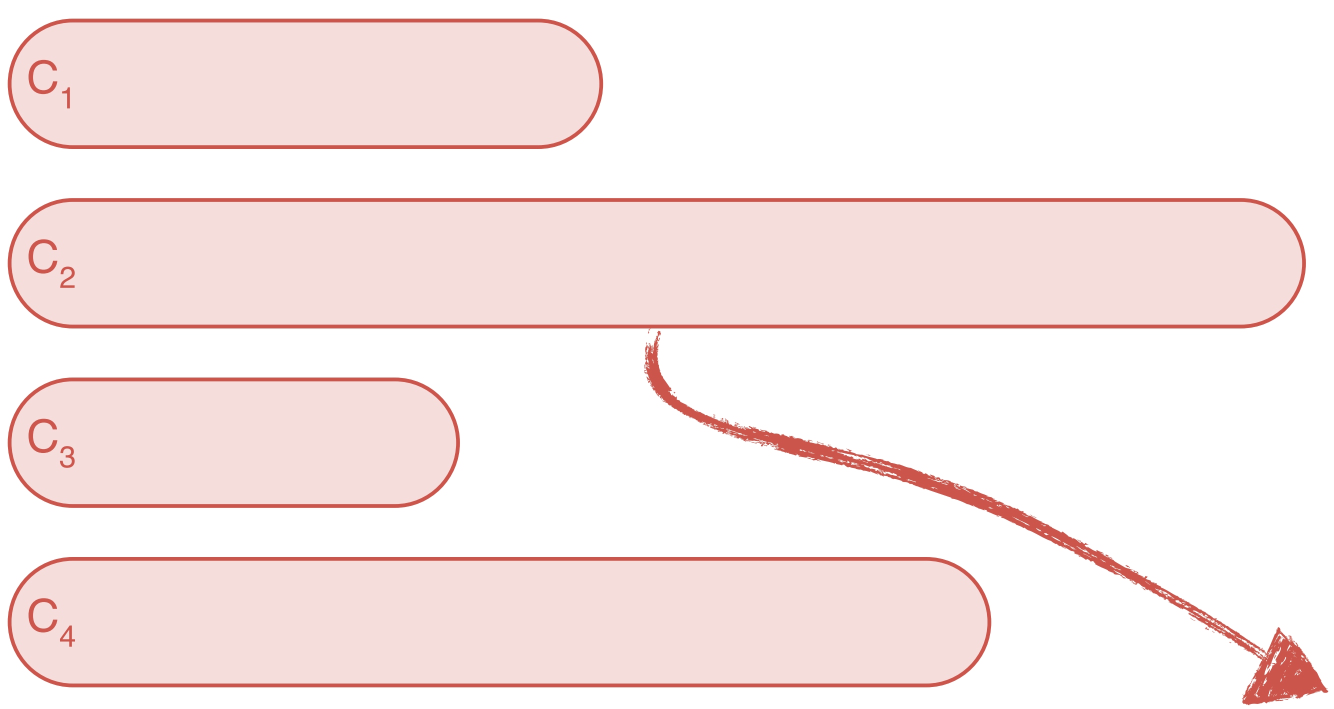
Theorem *Diagnosis* [Reiter'87,Prop 4.4]

$\Delta \subseteq$ COMPONENTS is a diagnosis for (SD, COMPONENTS,OBS) iff Δ is a minimal hitting set for the collection of (minimal) conflict sets for (SD, COMPONENTS,OBS).

Reiter's Theory of Diagnosis [Reiter'87]

Theorem *Diagnosis* [Reiter'87,Prop 4.4]

Δ⊆COMPONENTS is a diagnosis for (SD, COMPONENTS,OBS) iff Δ is a minimal hitting set for the collection of (minimal) conflict sets for (SD, COMPONENTS,OBS).

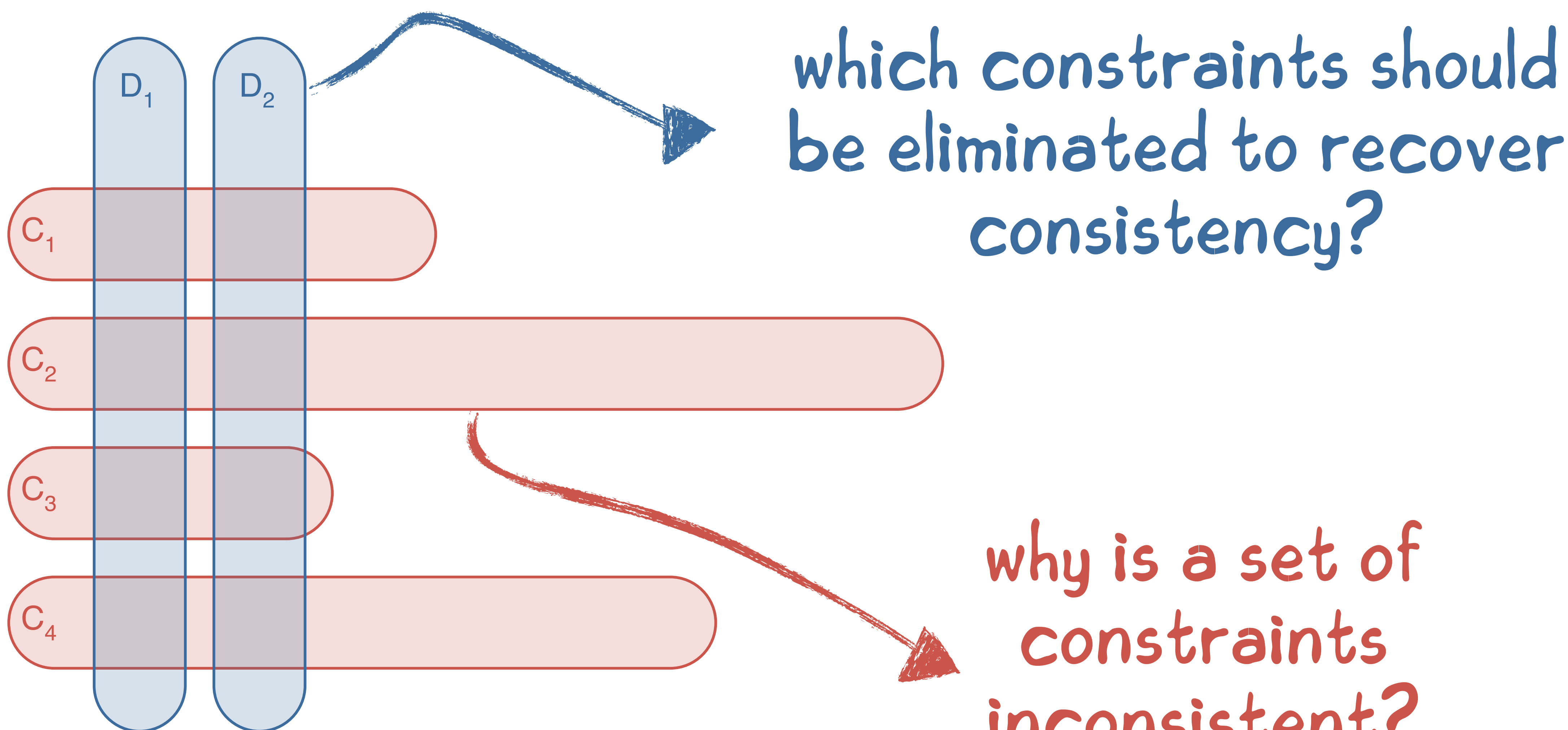


why is a set of constraints inconsistent?

Reiter's Theory of Diagnosis [Reiter'87]

Theorem *Diagnosis* [Reiter'87,Prop 4.4]

$\Delta \subseteq$ COMPONENTS is a diagnosis for (SD, COMPONENTS,OBS) iff Δ is a minimal hitting set for the collection of (minimal) conflict sets for (SD, COMPONENTS,OBS).



Reiter's Theory of Diagnosis [Reiter'87]

Example *Binary full adder* [Reiter'87]

The full adder has two minimal conflict sets $\{X_1, X_2\}$ and $\{X_1, A_2, O_1\}$ corresponding, respectively, to the inconsistency of:

$$SD \cup OBS \cup \{\neg AB(X_1), \neg AB(X_2)\}$$

$$SD \cup OBS \cup \{\neg AB(X_1), \neg AB(A_2), \neg AB(O_1)\}$$

There are three diagnoses: $\{X_1\}$, $\{X_2, A_2\}$, $\{X_2, O_1\}$

Reiter's Theory of Diagnosis [Reiter'87]

Definition *HS-tree* [Reiter'87,Def 4.6]

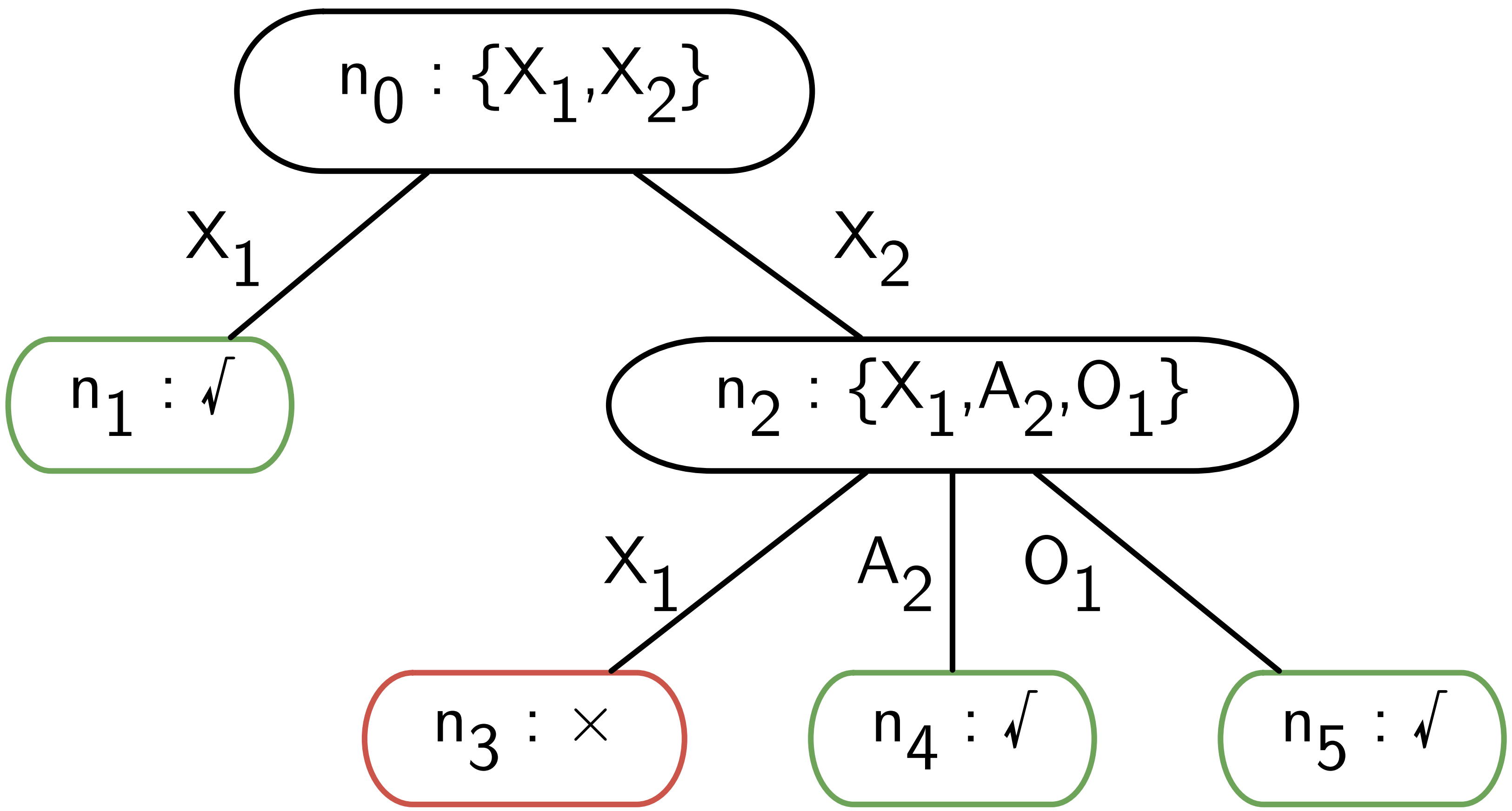
Let C be a collection of sets. An HS-tree for C, call it T, is a smallest edge-labeled and node-labeled tree with the following properties:

1. The root is labeled by $\surd$ if C is empty. Otherwise the root is labeled by an arbitrary set of C;
2. For each node n of T, let $H(n)$ be the set of edge labels on the path in T from the root node to n. The label for n is any set $S \in C$ such that $S \cap H(n) = \{\}$, if such a set S exists. Otherwise, the label for n is $\surd$. If n is labeled by the set S, then for each $\sigma \in S$, n has a successor, n_σ, joined to n by an edge labeled by σ.

Reiter's Theory of Diagnosis [Reiter'87]

Example *HS-Tree* [Reiser'87]

T



$C = \{\{X_1,X_2\},\{X_1,A_2,O_1\}\}$

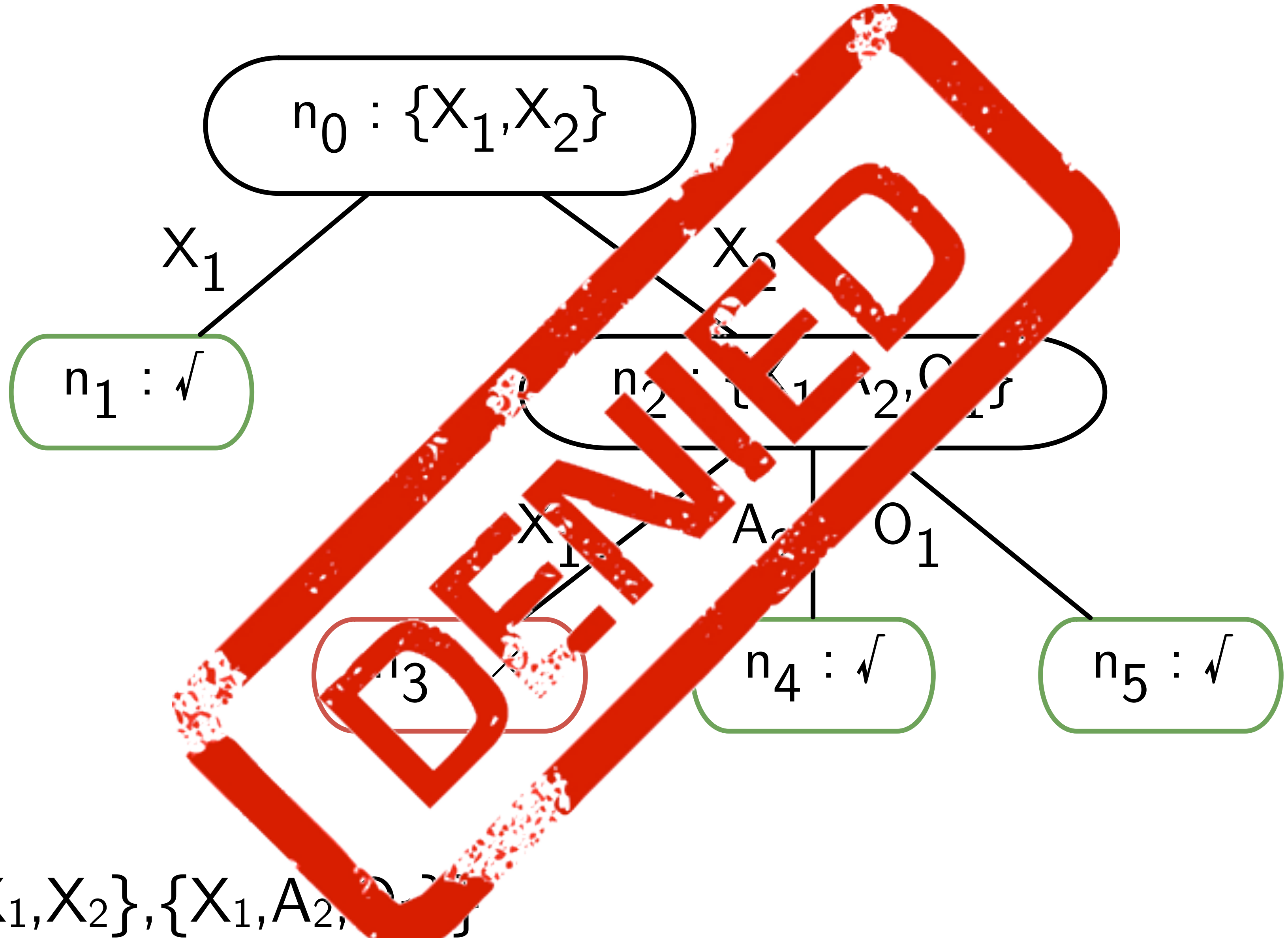
$label(n_0) = \{X_1,X_2\}$, $label(n_1) = \surd$, $label(n_2) = \{X_1,A_2,O_1\}$,...

$H(n_1) = \{X_1\}$, $H(n_4) = \{X_2,A_2\}$, $H(n_5) = \{X_2,O_1\}$,...

Reiter's Theory of Diagnosis [Reiter'87]

Example *HS-Tree* [Reiser'87]

T



$C = \{\{X_1,X_2\},\{X_1,A_2,O_1\}\}$

$label(n_0) = \{X_1,X_2\}$, $label(n_1) = \surd$, $label(n_2) = \{X_1,A_2,O_1\},\ldots$

$H(n_1) = \{X_1\}$, $H(n_4) = \{X_2,A_2\}$, $H(n_5) = \{X_2,O_1\},\ldots$

Reiter's Theory of Diagnosis [Reiter'87]

Algorithm *HS-DAG* [Greiner'89]

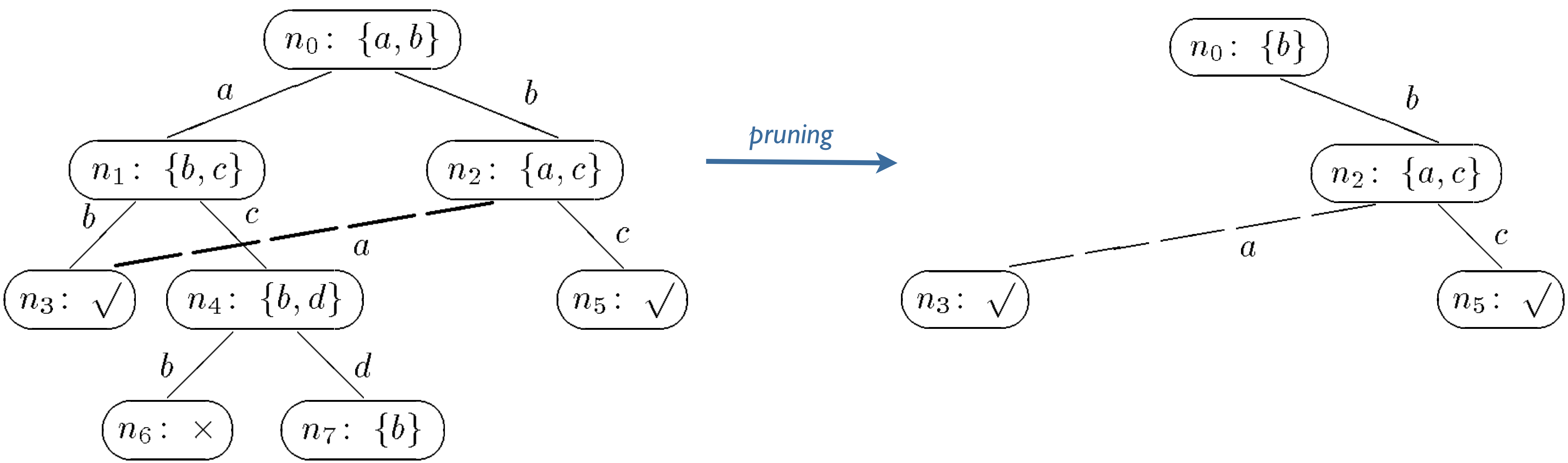
1. **let** D **be** the growing DAG.
   generate the root of the DAG
2. Process the nodes in D in breadth first order:
   A. **let** H(n) **be** the set of edge labels on the path from the root to n
   B. **if** $\forall x \in C, x \cap H(n) \neq \{\}$ **then** label n by √
      else label n by S **where** S is the first member of C for which $S \cap H(n) = \{\}$
   C. **if** n is labeled by a set $S \in C$
      then $\forall s \in S$, **generate** a new downward arc labeled by s. This arc leads to a new node m with $H(m) = H(n) \cup \{s\}$.

+ PRUNING

Reiter's Theory of Diagnosis [Reiter'87]

Example *HS-DAG* [Greiner'89]

D



C = {{a,b}, {b,c} {a,c},{a,b},{b,d},{b}}

H(n_3) = {a,b}, H(n_5) = {b,c}

Reiter's Theory of Diagnosis [Reiter'87]

Theorem *Correctness of HS-DAG algorithm* [Greiner'89, Th 2]

Given the ordered collection C, the HS-DAG algorithm returns a particular labeled DAG such that;

1. For all nodes labeled by $\surd$, H(n) is a minimal hitting set;
2. Every minimal hitting set for C is H(n) for some node n whose label is $\surd$.

Reiter's Theory of Diagnosis [Reiter'87]

Property *Diagnosis cardinality* [Reiner'87]

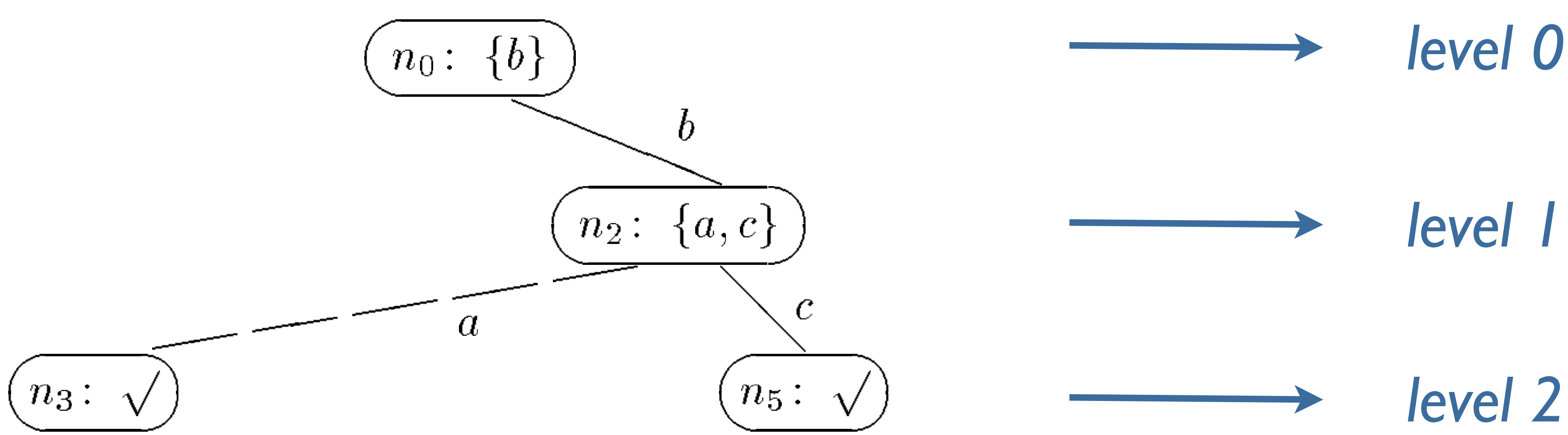
The breadth-first generation of the DAG results in the computation of diagnosis in order of increasing cardinality. If diagnosis of cardinality lower than k are desired, then HS-DAG can stop growing the DAG at level k.

Reiter's Theory of Diagnosis [Reiter'87]

Property *Diagnosis cardinality* [Reiner'87]

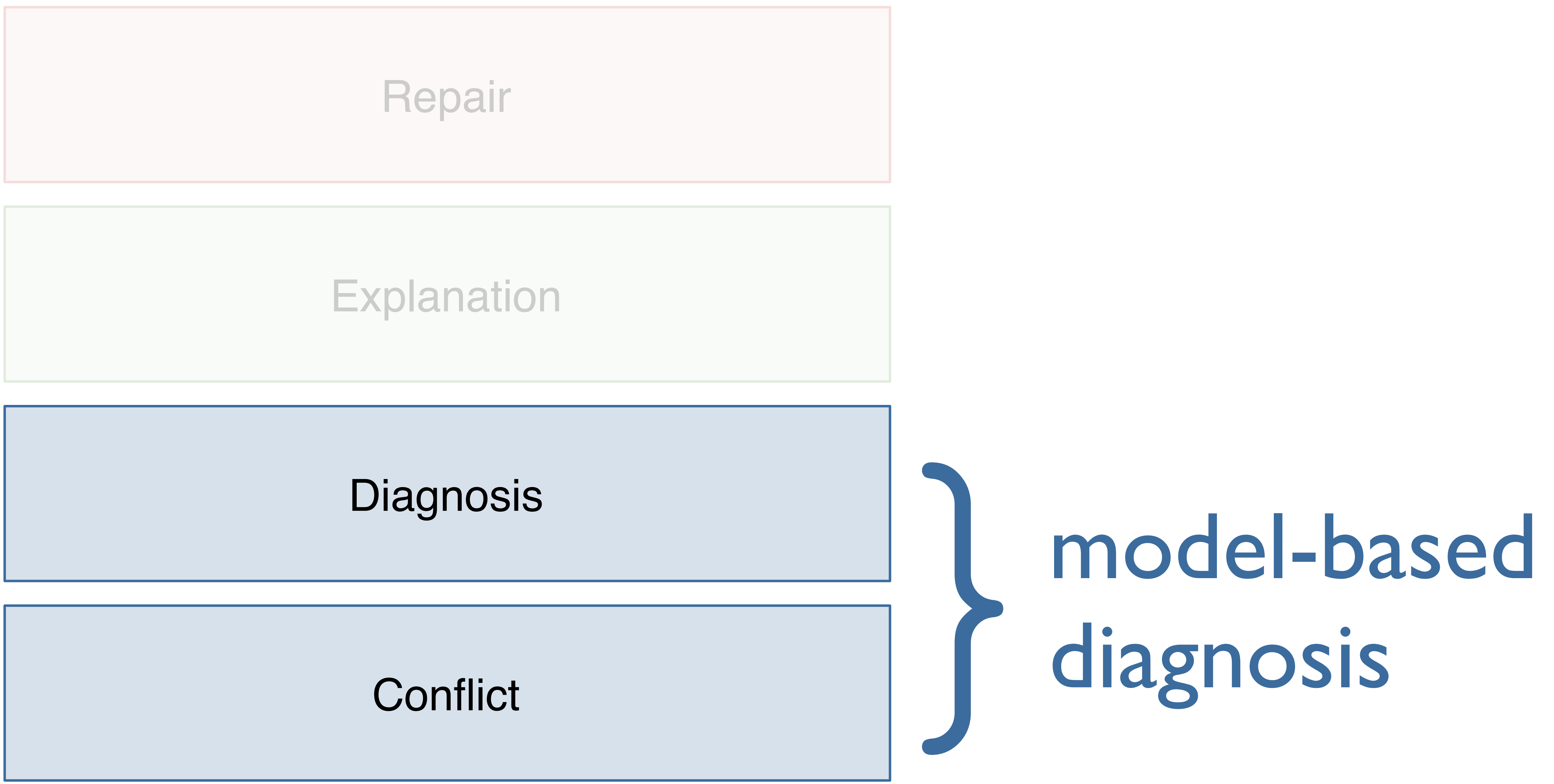
The breadth-first generation of the DAG results in the computation of diagnosis in order of increasing cardinality. If diagnosis of cardinality lower than k are desired, then HS-DAG can stop growing the DAG at level k.

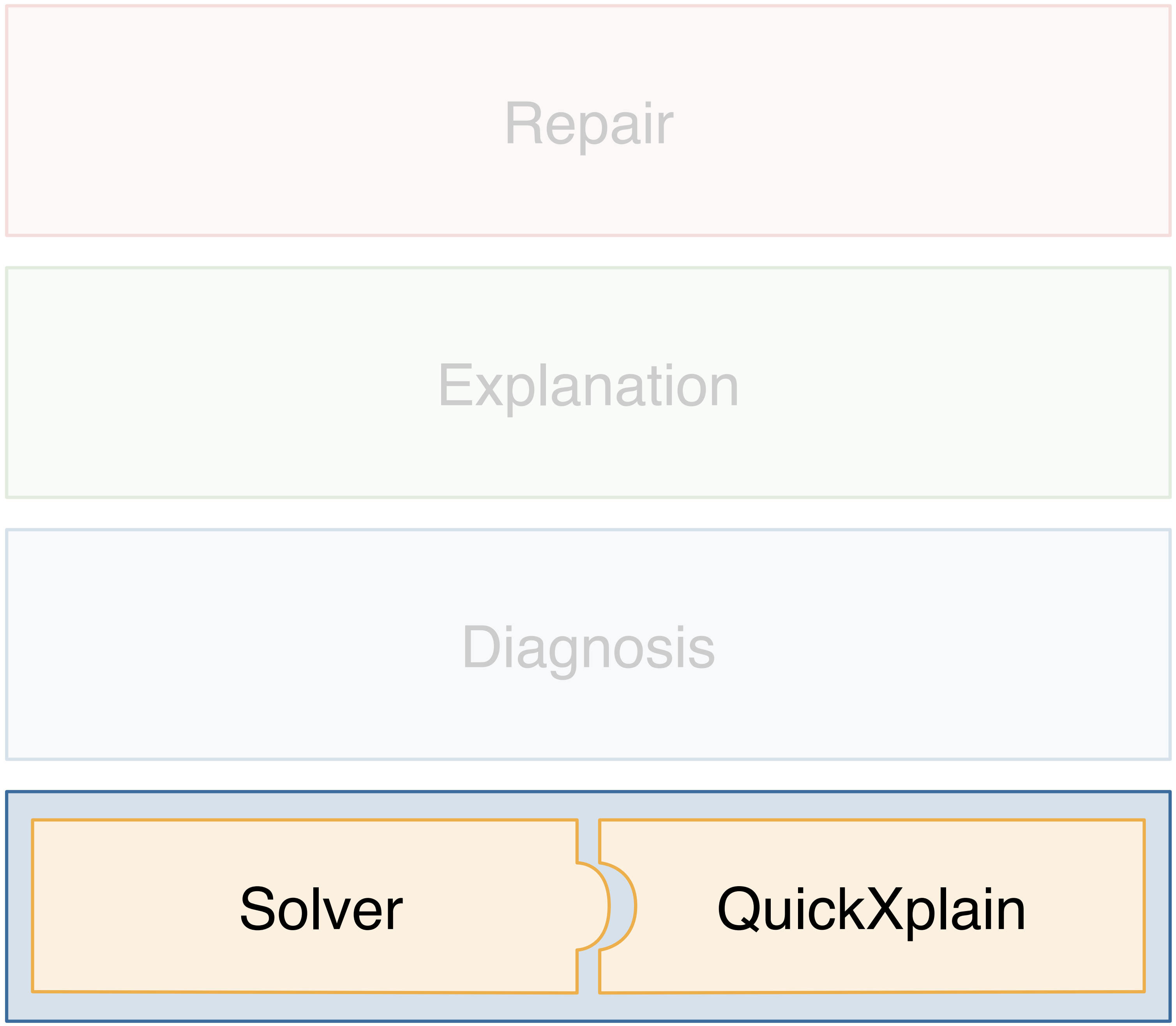
Example *HS-DAG* [Greiner'89]



Reiter's Theory of Diagnosis [Reiter'87]

- and much more:
  - min # calls to solver
  - single fault diagnosis
  - predictors
  - relationship with default logic
  - ...


Repair
Explanation
Diagnosis
Conflict
model-based
diagnosis


Repair
Explanation
Diagnosis
Solver
QuickXplain

QuickXplain [Junker'04]

- Complex problem:
  - exponential # conflicts: explain failure
  - exponential # relaxations: restore consistency
- Users of interactive applications want:
  - explanations for most important constraints
  - preference-based relaxation

QuickXplain [Junker'04]

Example *Preferred explanation* [Junker'04]

Example 1 *A customer wants to buy a station-wagon with following options, but has a limited budget of* 3000*:*

	Option	*Requirement* ρ_i	*Costs*
1.	*roof racks*	$x_1 = 1$	$k_1 = 500$
2.	*CD-player*	$x_2 = 1$	$k_2 = 500$
3.	*one additional seat*	$x_3 = 1$	$k_3 = 800$
4.	*metal color*	$x_4 = 1$	$k_4 = 500$
5.	*special luxury version*	$x_5 = 1$	$k_5 = 2600$

where the boolean variable $x_i \in \{0, 1\}$ *indicates whether the* i*-th option is chosen and the costs* $y = \sum_{i=1}^{5} k_i \cdot x_i$ *are smaller than the total budget of* 3000.

QuickXplain [Junker'04]

Example *Preferred explanation* [Junker'04]

Requirement	Deduction	Argument/Conflict
$\rho_1 : x_1 = 1$	$y \geq 500$	$\{\rho_1\}$
$\rho_2 : x_2 = 1$	$y \geq 1000$	$\{\rho_1, \rho_2\}$
$\rho_3 : x_3 = 1$	$y \geq 1800$	$\{\rho_1, \rho_2, \rho_3\}$
$\rho_4 : x_4 = 1$	$y \geq 2300$	$\{\rho_1, \rho_2, \rho_3, \rho_4\}$
$\rho_5 : x_5 = 1$	$y \geq 4900$	$\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5\}$
	$fail$	$\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5\}$

Table 1: Computing a conflict during propagation.

complete set of requirements

$\updownarrow$

minimal (irreductible) conflicts

QuickXplain [Junker'04]

Example — *Preferred explanation* [Junker'04]

Requirement	Deduction	Argument/Conflict
$\rho_4 : x_4 = 1$	$y \geq 500$	$\{\rho_4\}$
$\rho_5 : x_5 = 1$	$y \geq 3100$	$\{\rho_4, \rho_5\}$
	$fail$	$\{\rho_4, \rho_5\}$

Table 2: Propagation for producing a minimal conflict.

QuickXplain [Junker'04]

Example *Preferred explanation* [Junker'04]

Requirement	Deduction	Argument/Conflict
$\rho_4 : x_4 = 1$	$y \geq 500$	$\{\rho_4\}$
$\rho_5 : x_5 = 1$	$y \geq 3100$	$\{\rho_4, \rho_5\}$
	$fail$	$\{\rho_4, \rho_5\}$

Table 2: Propagation for producing a minimal conflict.

Requirement	Deduction	Argument/Conflict
$\rho_3 : x_3 = 1$	$y \geq 800$	$\{\rho_3\}$
$\rho_5 : x_5 = 1$	$y \geq 3400$	$\{\rho_3, \rho_5\}$
	$fail$	$\{\rho_3, \rho_5\}$

Table 3: A preferred explanation for $\rho_3 \prec \rho_1 \prec \rho_2 \prec \rho_5 \prec \rho_4$.

QuickXplain [Junker'04]

Definition *Problem* [Junker'04]

A problem P is defined as a pair (B,C) where:

1. B is the set of background constraints. Constraints in B cannot be relaxed;
2. C is the set of user requirements. Constraints in C can be relaxed.

QuickXplain [Junker'04]

Definition *Problem* [Junker'04]

A problem P is defined as a pair (B,C) where:

1. B is the set of background constraints. Constraints in B cannot be relaxed;
2. C is the set of user requirements. Constraints in C can be relaxed.

Note *Relation with Reiter's (SD,COMPONENTS,OBS)*

B ≃ SD (system description)

C ≃ OBS (observation)

variables ≃ COMPONENTS

QuickXplain [Junker'04]

Definition *Relaxation of a problem* [Junker'04, Def 1]

A set $R \subseteq C$ is a *relaxation* of a problem $P=(B,C)$ iff $B \cup R$ has **a** solution.

Definition *Conflict of a problem* [Junker'04, Def 1]

A set $F \subseteq C$ is a *conflict* of a problem $P=(B,C)$ iff $B \cup F$ has **no** solution.

QuickXplain [Junker'04]

Example — *Relaxation vs Conflict* [Junker'04]

Example 1 *A customer wants to buy a station-wagon with following options, but has a limited budget of* 3000*:*

	Option	*Requirement* ρ_i	*Costs*
1.	*roof racks*	$x_1 = 1$	$k_1 = 500$
2.	*CD-player*	$x_2 = 1$	$k_2 = 500$
3.	*one additional seat*	$x_3 = 1$	$k_3 = 800$
4.	*metal color*	$x_4 = 1$	$k_4 = 500$
5.	*special luxury version*	$x_5 = 1$	$k_5 = 2600$

where the boolean variable $x_i \in \{0,1\}$ *indicates whether the* i*-th option is chosen and the costs* $y = \sum_{i=1}^{5} k_i \cdot x_i$ *are smaller than the total budget of* 3000.

B: $y \leq 3000$

C: $x_1=1$, $x_2=1$, $x_3=1$, $x_4=1$, $x_5=1$

QuickXplain [Junker'04]

Example *Relaxation vs Conflict* [Junker'04]

B: $y \leq 3000$

C: $x_1=1,\ x_2=1,\ x_3=1,\ x_4=1,\ x_5=1$

Requirement	Deduction	Argument/Conflict
$\rho_1 : x_1 = 1$	$y \geq 500$	$\{\rho_1\}$
$\rho_2 : x_2 = 1$	$y \geq 1000$	$\{\rho_1, \rho_2\}$
$\rho_3 : x_3 = 1$	$y \geq 1800$	$\{\rho_1, \rho_2, \rho_3\}$
$\rho_4 : x_4 = 1$	$y \geq 2300$	$\{\rho_1, \rho_2, \rho_3, \rho_4\}$
$\rho_5 : x_5 = 1$	$y \geq 4900$	$\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5\}$
	$fail$	$\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5\}$

Table 1: Computing a conflict during propagation.

F: $\{x_1=1,\ x_2=1,\ x_3=1,\ x_4=1,\ x_5=1\}$

R: $\{\}$

QuickXplain [Junker'04]

Definition *Maximal relaxation of a problem* [Junker'04]

A relaxation $R \subseteq C$ is *maximal* (non-extensible) iff there is no other relaxation $R' \supseteq R$ s.t. $B \cup R'$ has a solution.

Definition *Minimal conflict of a problem* [Junker'04]

A conflict $F \subseteq C$ is *minimal* (irreductible) iff for every $F' \subseteq F$ $B \cup F'$ has a solution.

QuickXplain [Junker'04]

Example *Maximal relaxation vs Minimal conflict* [Junker'04]

B: $y \leq 3000$

C: $x_1=1$, $x_2=1$, $x_3=1$, $x_4=1$, $x_5=1$

Requirement	Deduction	Argument/Conflict
$\rho_4 : x_4 = 1$	$y \geq 500$	$\{\rho_4\}$
$\rho_5 : x_5 = 1$	$y \geq 3100$	$\{\rho_4, \rho_5\}$
	$fail$	$\{\rho_4, \rho_5\}$

Table 2: Propagation for producing a minimal conflict.

F: $\{x_4=1, x_5=1\}$

R: $\{x_1=1, x_2=1, x_3=1\}$

QuickXplain [Junker'04]

Definition *Preference relation* [Junker'04]

The strict partial order between the constraints of C, denoted by $<$, defines a *preference relation* between constraints. We write $c_1 < c_2$ iff the selection of c_1 is preferred to the selection of c_2.

QuickXplain [Junker'04]

Definition *Preference relation* [Junker'04]

The strict partial order between the constraints of C, denoted by $\prec$, defines a *preference relation* between constraints. We write $c_1 \prec c_2$ iff the selection of c_1 is preferred to the selection of c_2.

Definition *Linearization* [Junker'04]

A *linearization* $<$ of $\prec$ is a strict total order that is a superset of $\prec$ and which describes a complete ranking.

QuickXplain [Junker'04]

Definition *Preferred relaxation over a strict total order* [Junker'04, Def 4]

Let $P=(B,C,<)$ be a totally ordered problem. A relaxation R of P is a *preferred relaxation* of P iff there is no other relaxation R' of P s.t. $R' <_{lex} R$.

has more preferred constrains

QuickXplain [Junker'04]

Definition *Preferred relaxation over a strict total order* [Junker'04, Def 4]

Let $P=(B,C,<)$ be a totally ordered problem. A relaxation R of P is a *preferred relaxation* of P iff there is no other relaxation R' of P s.t. $R' <_{lex} R$.

has more preferred constrains

Definition *Preferred relaxation over a strict partial order* [Junker'04, Def 5]

Let $P=(B,C,\prec)$ be a partially ordered problem. A relaxation R of P is a *preferred relaxation* of P iff there is a linearization $<$ of $\prec$ s.t. R is a preferred relaxation of $(B,C,<)$.

QuickXplain [Junker'04]

Definition *Preferred conflict over a strict total order* [Junker'04, Def 7]

Let P=(B,C,<) be a totally ordered problem. A conflict F of P is a *preferred conflict* of P iff there is no other relaxation F' of P s.t. $F' <_{antilex} F$.

has less preferred constrains

QuickXplain [Junker'04]

Definition *Preferred conflict over a strict total order* [Junker'04, Def 7]

Let P=(B,C,<) be a totally ordered problem. A conflict F of P is a *preferred conflict* of P iff there is no other relaxation F' of P s.t. $F' <_{antilex} F$.

has less preferred constrains

Definition *Preferred conflict over a strict partial order* [Junker'04, Def 8]

Let P=(B,C,$\prec$) be a partially ordered problem. A conflict F of P is a *preferred conflict* of P iff there is a linearization < of $\prec$ s.t. F is a preferred conflict of (B,C,<).

QuickXplain [Junker'04]

Example *Maximal relaxation vs Minimal conflict* [Junker'04]

B: $y \leq 3000$

C: $x_1=1$, $x_2=1$, $x_3=1$, $x_4=1$, $x_5=1$

Requirement	Deduction	Argument/Conflict
$\rho_3 : x_3 = 1$	$y \geq 800$	$\{\rho_3\}$
$\rho_5 : x_5 = 1$	$y \geq 3400$	$\{\rho_3, \rho_5\}$
	$fail$	$\{\rho_3, \rho_5\}$

Table 3: A preferred explanation for $\rho_3 \prec \rho_1 \prec \rho_2 \prec \rho_5 \prec \rho_4$.

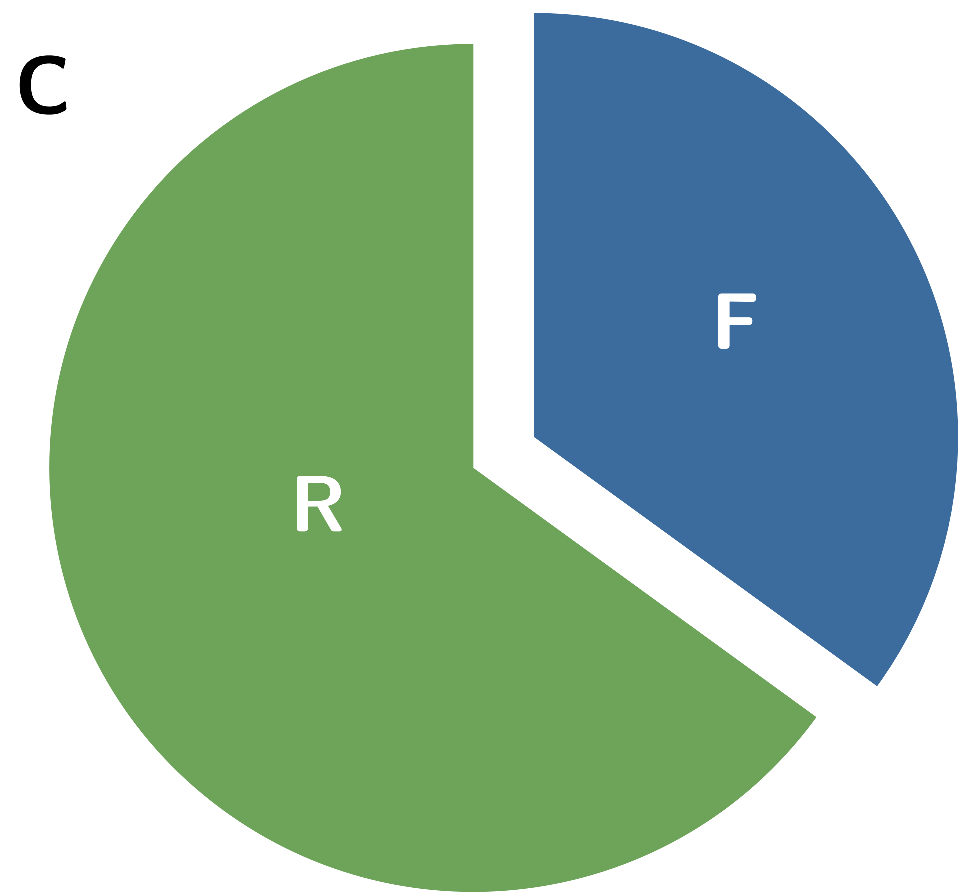
F: $\{x_3=1, x_5=1\}$

R: $\{x_1=1, x_2=1, x_4=1,\}$

QuickXplain [Junker'04]

Proposition *Preference duality* [Junker'04, Prop 4]

If F is a preferred conflict of P=(B,C,<) and R is a preferred relaxation of P, then the <-minimal element of C-R is equal to the <-maximal elements of F.



QuickXplain [Junker'04]

Algorithm *QuickXplain* [Junker'04]

QuickXplain($B,C,<$)

1. **if** isConsistent($B \cup C$) **return** "no conflict"
2. **else if** $C=\emptyset$ **then return** $\emptyset$
3. **else return** QuickXplain'($B,B,C,<$)

QuickXplain'($B,\Delta,C,<$)

4. **if** $\Delta \neq \emptyset$ **and** $\neg$isConsistent(B) **then return** $\emptyset$
5. **if** $C = \{c\}$ **then return** $\{c\}$
6. **let** $c_1,\ldots, c_n$ **be** *an enumeration of C that respects* $<$
7. **let** k **be** split(n) *where* $1 \leq k \leq n$
8. $C_1=\{c_1,\ldots, c_k\}$ **and** $C_2=\{c_{k+1},\ldots, c_n\}$
9. $\Delta_2=$ QuickXplain'($B \cup C_1,C_1,C_2,<$)
10. $\Delta_1=$ QuickXplain'($B \cup \Delta_2,\Delta_2,C_1,<$)
11. **return** $\Delta_1 \cup \Delta_2$

QuickXplain [Junker'04]

Theorem *QuickXplain's termination* [Junker'04, Th 1]

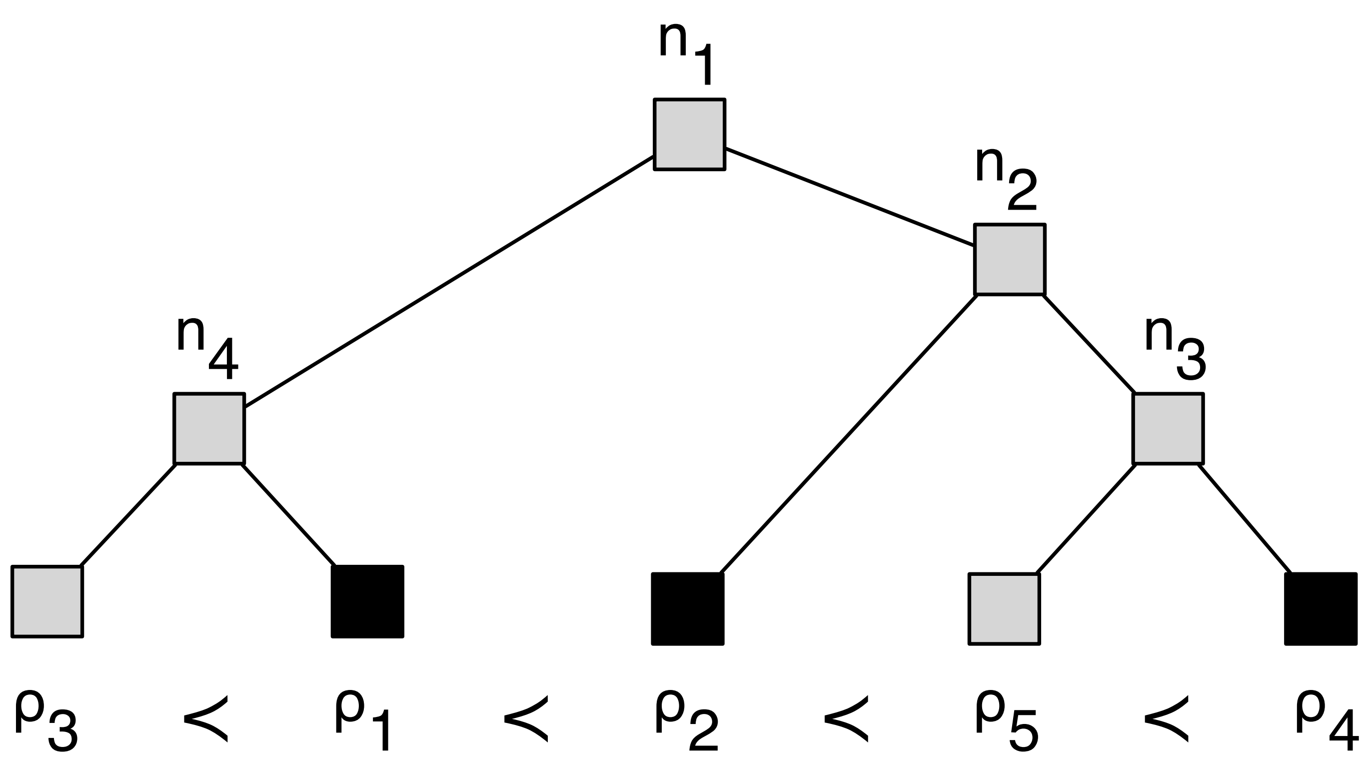
The algorithm QuickXplain(B,C,<) always terminates. If B∪C has a solution then it returns "no conflict". Otherwise, it returns a preferred conflict of (B,C, <).

QuickXplain [Junker'04]

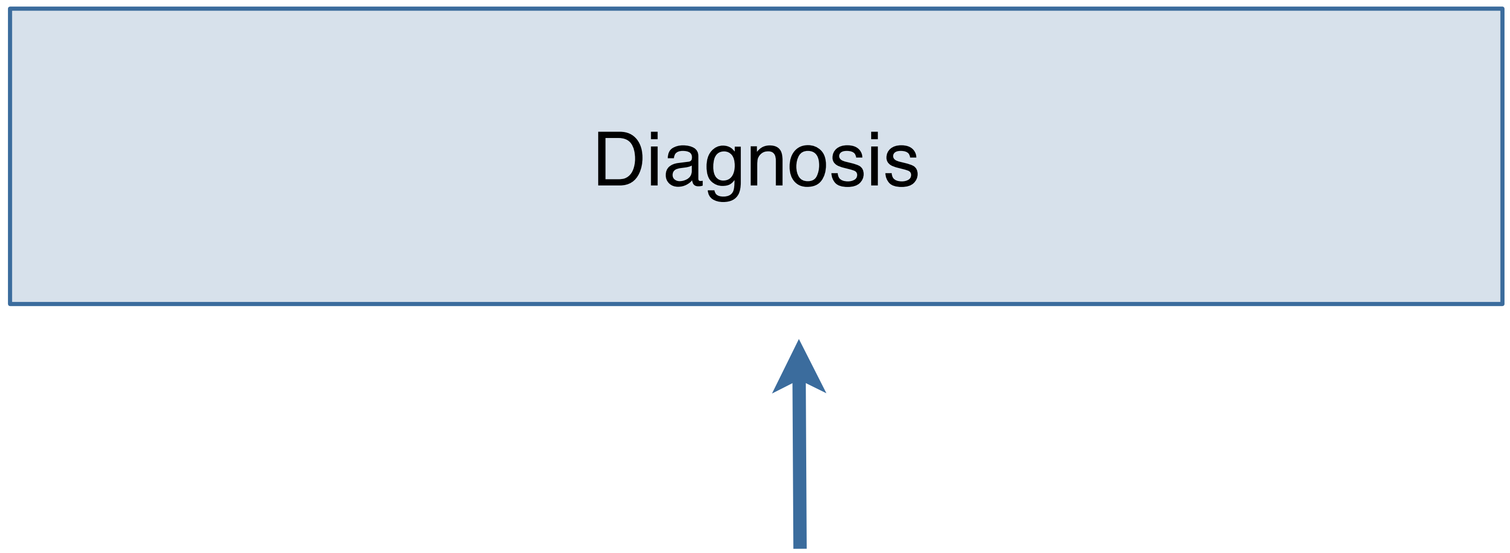
Example *Call graph of QuickXplain algorithm* [Junker'04]

Requirement	Deduction	Argument/Conflict
$\rho_3 : x_3 = 1$	$y \geq 800$	$\{\rho_3\}$
$\rho_5 : x_5 = 1$	$y \geq 3400$	$\{\rho_3, \rho_5\}$
	$fail$	$\{\rho_3, \rho_5\}$

Table 3: A preferred explanation for $\rho_3 \prec \rho_1 \prec \rho_2 \prec \rho_5 \prec \rho_4$.

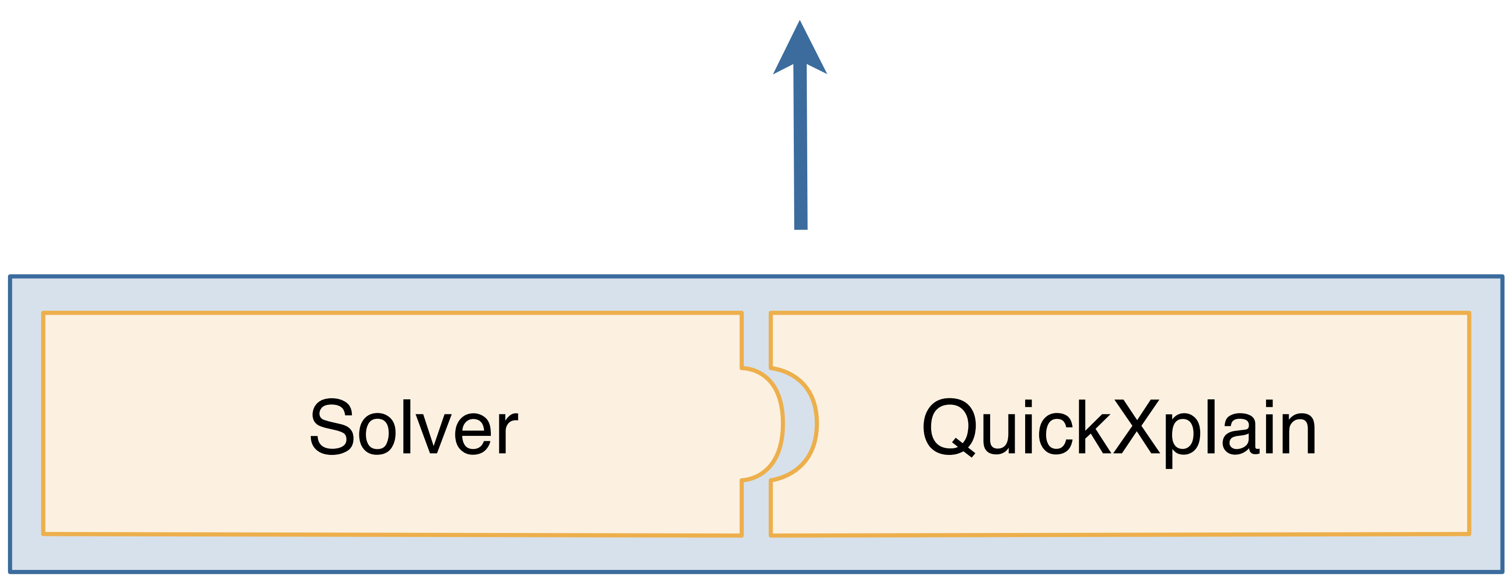


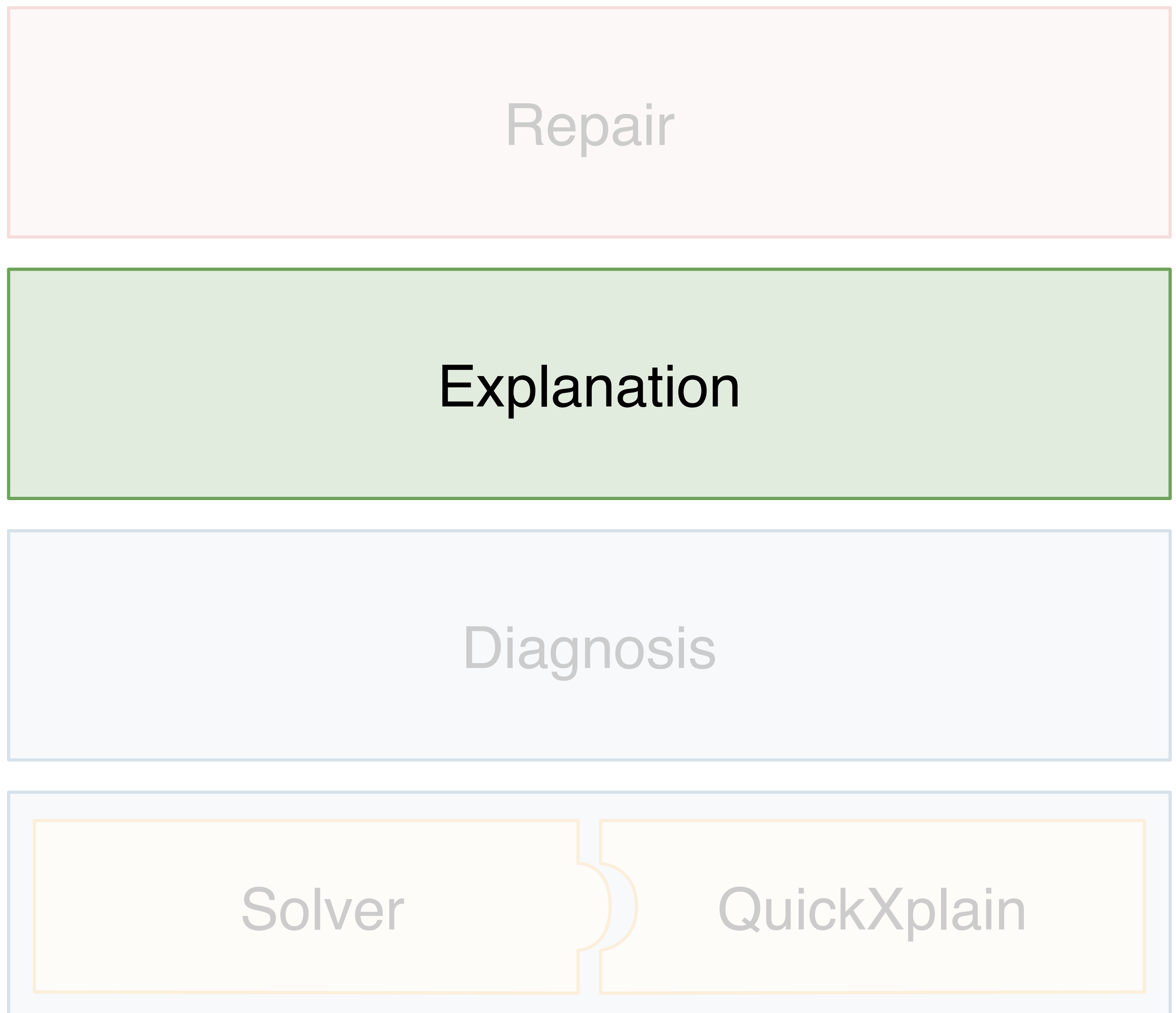
QuickXplain in Reiter's theory



Property *Minimal conflict set* [Reiter'87]

If the solver returns only minimal conflict sets, no edge will ever be marked redundant, which simplifies the generation algorithm.




Repair
Explanation
Diagnosis
Solver
QuickXplain

Optimal relaxation [Jannach'06]

- Research question:
  - How to efficiently find customer-optimal relaxations for a given recommendation problem?
- Approach:
  - model-based diagnosis
  - best-effort search algorithm

Optimal relaxation [Jannach'06]

Example *Camera recommender system* [Jannach'06]

P = product database

ID	USB	Firewire	Price	Resolution	Make
p1	true	false	400	5	Canon
p2	false	true	500	5	Canon
p3	true	false	200	4	Fuji
p4	false	true	400	5	HP

FRS = filter rules

f1: IF *the customer requires high-quality printouts of the pictures* THEN
recommend cameras with a resolution of 5 mega-pixels.

f2: IF *the customer wants to have a cheap camera* THEN
recommend cameras with a price smaller than 300.

f3: IF *customer needs a USB port* THEN
recommend cameras with a USB port.

f4: IF *customer wants extended connectivity* THEN
recommend cameras supporting Firewire.

f5: IN ANY CASE
recommend cameras with a resolution higher than 3 mega-pixels.

Optimal relaxation [Jannach'06]

Example *Camera recommender system* [Jannach'06]

FRS = filter rules

f1: IF *the customer requires high-quality printouts of the pictures* THEN
recommend cameras with a resolution of 5 mega-pixels.
f2: IF *the customer wants to have a cheap camera* THEN
recommend cameras with a price smaller than 300.
f3: IF *customer needs a USB port* THEN
recommend cameras with a USB port.
f4: IF *customer wants extended connectivity* THEN
recommend cameras supporting Firewire.
f5: IN ANY CASE
recommend cameras with a resolution higher than 3 mega-pixels.

CRS = customer requirements

USB-support, extended-connectivity, only cheap cameras, high-quality printouts.

Optimal relaxation [Jannach'06]

Definition *Optimal relaxation* [Jannach'06]

Let:

- RC be a function describing the costs of retracting a single filter f∈FRS;
- COSTS be a function describing the costs of a relaxation.

A relaxation R for a recommendation problem RP= (P,FRS,CRS) is optimal iff there does not exist R' such that:

$$\text{COSTS}(R',CRS,RC) < \text{COSTS}(R,CRS,RC)$$

Representative explanation [O'Sullivan]

- Research question:
  - How to define and compute representative explanations?
- Approach:
  - model-based diagnosis
  - definition of representativeness

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

	Option	*Selector*	*Cost*
c_1	*Budget*	$x_1 = 1$	$\sum_{i \in \{2,\ldots,5\}} (k_i \cdot x_i) \leq 3000$
c_2	*Roof Rack*	$x_2 = 1$	$k_2 = 500$
c_3	*Convertible*	$x_3 = 1$	$k_3 = 500$
c_4	*CD Player*	$x_4 = 1$	$k_4 = 500$
c_5	*Leather Seats*	$x_5 = 1$	$k_5 = 2600$

Assume that the technical constraints of the configuration problem forbid convertible cars having roof racks, therefore, constraints c_2 and c_3 form a conflict. Note that, given the budget constraint, if the user selects option c_5, it is not possible to have any of the options c_2, c_3, c_4. ▲

Minimal conflicts = $\{\{c_2,c_3\}, \{c_1,c_2,c_5\}, \{c_1,c_3,c_5\}, \{c_1,c_4,c_5\}\}$

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Table 1: The set of relaxations and exclusion sets for the over-constrained problem presented in Example 1. We show both the subset of the constraints in the relaxation (marked with a ✓) and those that are in the exclusion set, i.e. those that must be removed (marked with a ×).

Exp.	Constraints c_1	c_2	c_3	c_4	c_5	Relaxation	Exclusion Set
I	×	×	✓	✓	✓	$\{c_3, c_4, c_5\}$	$\{c_1, c_2\}$
II	×	✓	×	✓	✓	$\{c_2, c_4, c_5\}$	$\{c_1, c_3\}$
III	✓	×	✓	✓	×	$\{c_1, c_3, c_4\}$	$\{c_2, c_5\}$
IV	✓	✓	×	✓	×	$\{c_1, c_2, c_4\}$	$\{c_3, c_5\}$
V	✓	×	×	×	✓	$\{c_1, c_5\}$	$\{c_2, c_3, c_4\}$

Exponential number of relaxation/exclusion sets!

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 1: *every constraint that appears in a relaxation appears at least once in a relaxation in our chosen subset.*

Table 2: A set of representative relaxations.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	$\times$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
II	$\times$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
III	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 1: *every constraint that appears in a relaxation appears at least once in a relaxation in our chosen subset.*

Table 2: A set of representative relaxations.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
II	×	✓	×	✓	✓
III	✓	×	✓	✓	×

never excluded?

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 1: *every constraint that appears in a relaxation appears at least once in a relaxation in our chosen subset.*

Table 2: A set of representative relaxations.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
II	×	✓	×	✓	✓
III	✓	×	✓	✓	×

never c_1 and c_5?

never excluded?

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 1: *every constraint that appears in a relaxation appears at least once in a relaxation in our chosen subset.*

Table 2: A set of representative relaxations.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
II	×	✓	×	✓	✓
III	✓	×	✓	✓	×

never c_1 and c_5?

never excluded?

Explanation V

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 2: *every constraint that must be relaxed once, appears in at least one exclusion set.*

Table 3: A set of representative exclusion sets.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	$\times$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
III	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$
V	$\checkmark$	$\times$	$\times$	$\times$	$\checkmark$

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 2: *every constraint that must be relaxed once, appears in at least one exclusion set.*

Table 3: A set of representative exclusion sets.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	$\times$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
III	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$
V	$\checkmark$	$\times$	$\times$	$\times$	$\checkmark$

always excluded?

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 2: *every constraint that must be relaxed once, appears in at least one exclusion set.*

Table 3: A set of representative exclusion sets.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	$\times$	$\times$	✓	✓	✓
III	✓	$\times$	✓	✓	$\times$
V	✓	$\times$	$\times$	$\times$	✓

always excluded? → Explanations II, IV

Representative explanation [O'Sullivan'07]

Example *Car configuration problem* [O'Sullivan'07]

Option 3: *set of explanations should contain at least:*

1. *one maximal relaxation containing each satisfiable constraint*
2. *one minimal exclusion containing each excludable constraint*

Table 4: A set of representative explanations.

Explanation	Constraints c_1	c_2	c_3	c_4	c_5
I	$\times$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
IV	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\times$
V	$\checkmark$	$\times$	$\times$	$\times$	$\checkmark$

Representative explanation [O'Sullivan'07]

Definition *Maximal relaxation set* [O'Sullivan'07]

Let ρ be the set of all *maximal relaxations* of a problem $P=(B,C)$. Every pair of maximal relaxations are set-wise incomparable:

$$\forall R_i, R_j \in \rho : i \neq j \wedge R_i \not\subseteq R_j \wedge R_i \not\supseteq R_j$$

Representative explanation [O'Sullivan'07]

Definition *Maximal relaxation set* [O'Sullivan'07]

Let ρ be the set of all *maximal relaxations* of a problem $P=(B,C)$. Every pair of maximal relaxations are set-wise incomparable:

$$\forall R_i, R_j \in \rho : i \neq j \wedge R_i \not\subseteq R_j \wedge R_i \not\supseteq R_j$$

Definition *Minimal exclusion (conflict) set* [O'Sullivan'07]

Let Φ be the set of all *minimal exclusions* (conflicts) of a problem $P=(B,C)$. Every pair of minimal exclusions are set-wise incomparable:

$$\forall C_i, C_j \in \Phi : i \neq j \wedge C_i \not\subseteq C_j \wedge C_i \not\supseteq C_j$$

Representative explanation [O'Sullivan'07]

Definition *Representative set of explanations* [O'Sullivan'07]

Given a problem $P=(B,C)$ that is inconsistent, $\rho' \subseteq \rho$ a set of maximal relaxations of P, and $\Phi'=\{C-R \mid R \in \rho'\}$ the corresponding set of minimal exclusion sets. The set:

$$X=\{(R,C-R) \mid R \in \rho'\}$$

is a *representative set* of explanations iff:

1. $\forall c \in \bigcup_{R \in \rho} : \exists R \in \rho' : c \in R$

2. $\forall c \in \bigcup_{C \in \Phi} : \exists C \in \Phi' : c \in C$

Representative explanation [O'Sullivan'07]

Definition *Minimal representative set of explanations* [O'Sullivan'07]

Given a problem P=(B,C) that is inconsistent, a set of representative set of explanations is minimal if any strict subset of it is not representative.

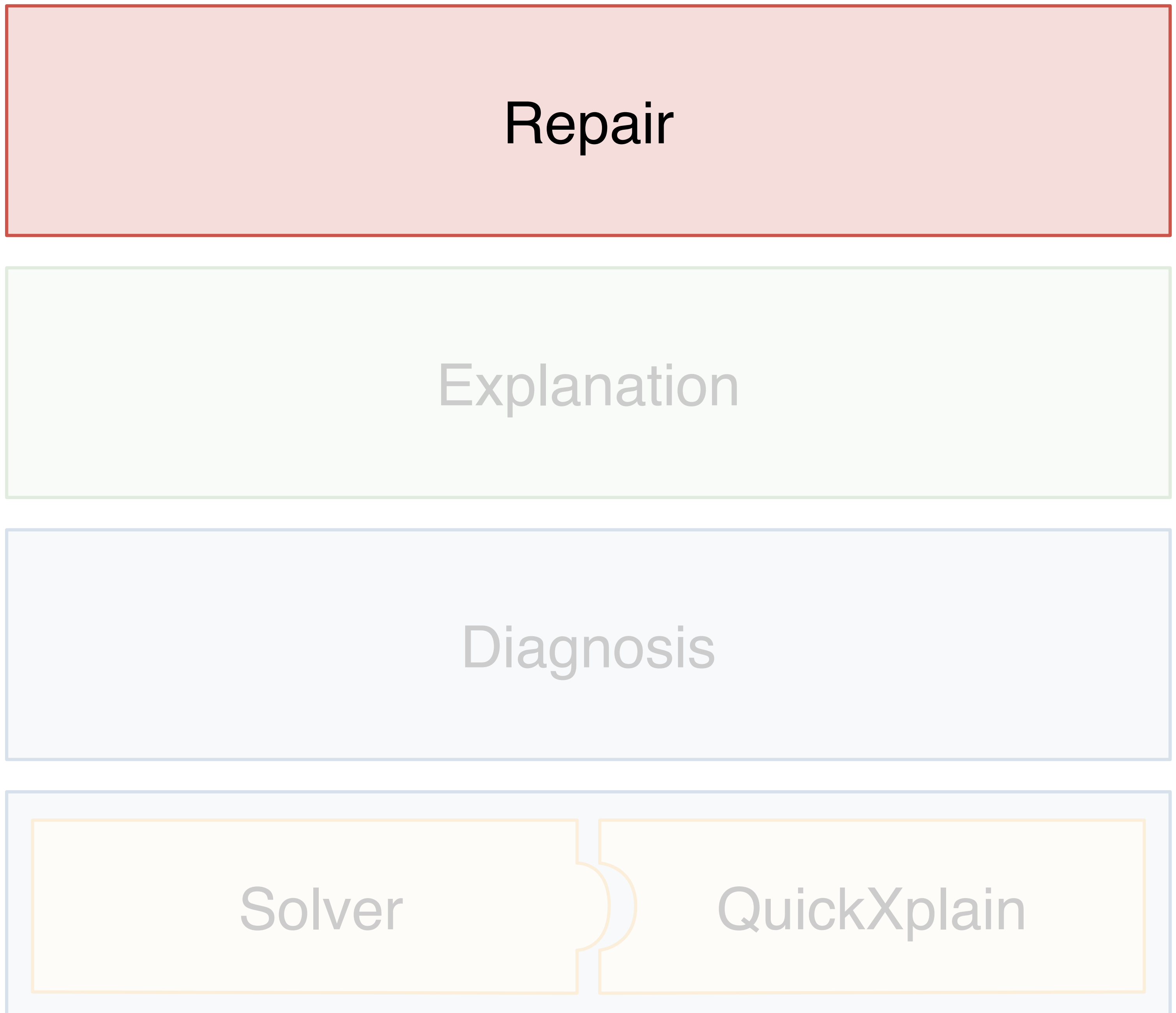
Representative explanation [O'Sullivan'07]

Definition *Minimal representative set of explanations* [O'Sullivan'07]

Given a problem P=(B,C) that is inconsistent, a set of representative set of explanations is minimal if any strict subset of it is not representative.

Theorem *Number of explanations* [O'Sullivan'07]

The size of any minimal representative set of explanations is at most |C|, and this bound is tight.


Repair
Explanation
Diagnosis
Solver
QuickXplain

Personalized repairs [Felfernig'09]

- Research question:
  - How to present a personalized repair approach that integrates the calculation of explanations with collaborative problem solving techniques?
- Approach:
  - model-based diagnosis
  - collaborative problem solving (k-nearest neighbour algorithms)

Personalized repairs [Felfernig'09]

Example *Financial services* [Felfernig'09]

Assumption: available knowledge-base of products

id	return-rate (p.a.)	run-time	risk level	shares percentage	accessibility	plow back earnings	blue chip
i_1	4.2	3.0	A	0	no	yes	yes
i_2	4.7	3.5	B	10	yes	no	yes
i_3	4.8	3.5	A	10	yes	yes	yes
i_4	5.2	4.0	B	20	no	no	no
i_5	4.3	3.5	A	0	yes	yes	yes
i_6	5.6	5.0	C	30	no	yes	no
i_7	6.7	6.0	C	50	yes	no	no
i_8	7.9	7.0	C	50	no	no	no

$C = \{\text{return-rate}\geq 5.5(r_1), \text{runtime}=3(r_2), \text{accessibility}=y(r_3), \text{bluechip}=y(r_4)\}$

Minimal diagnosis: $\{\{r_1,r_2\}, \{r_1,r_3\}, \{r_2,r_4\}\}$

Personalized repairs [Felfernig'09]

Example *Financial services* [Felfernig'09]

Minimal diagnosis: $\{\{r_1,r_2\}, \{r_1,r_3\}, \{r_2,r_4\}\}$

$R = \{\text{accessibility}=y(r_3),\ \text{bluechip}=y(r_4)\} \longrightarrow$ **rep_1, rep_2, rep_3**

$R = \{\text{runtime}=3(r_2),\ \text{bluechip}=y(r_4)\} \longrightarrow$ **rep_4**

$R = \{\text{return-rate}\geq 5.5(r_1),\ \text{accessibility}=y(r_3)\} \longrightarrow$ **rep_5**

repair	return-rate	runtime	accessibility	bluechip
rep_1	4.7	3.5	√	√
rep_2	4.8	3.5	√	√
rep_3	4.3	3.5	√	√
rep_4	4.2	√	no	√
rep_5	√	6.0	√	no

repair alternatives

Personalized repairs [Felfernig'09]

Definition *Similarity function* [Felfernig'09]

Given a problem $P=(B,C)$, the similarity function returns for every item (product) i_j the degree of similarity with the user requirements C.

Personalized repairs [Felfernig'09]

Definition *Similarity function* [Felfernig'09]

Given a problem P=(B,C), the similarity function returns for every item (product) i_j the degree of similarity with the user requirements C.

Example *Similarity function* [Felfernig'09]

$$sim(C, i_j) = \sum_{k=1}^{m} \left(1 - \frac{|c_k - i_j[k]|}{\max(k) - \min(k)}\right) * w(k)$$

Personalized repairs [Felfernig'09]

Example *Financial services* [Felfernig'09]

id	return-rate (p.a.)	run-time	risk level	shares percen-tage	accessi-bility	plow back earnings	blue chip
i_1	4.2	3.0	A	0	no	yes	yes
i_2	4.7	3.5	B	10	yes	no	yes
i_3	4.8	3.5	A	10	yes	yes	yes
i_4	5.2	4.0	B	20	no	no	no
i_5	4.3	3.5	A	0	yes	yes	yes
i_6	5.6	5.0	C	30	no	yes	no
i_7	6.7	6.0	C	50	yes	no	no
i_8	7.9	7.0	C	50	no	no	no

C: return-rate≥5.5(r_1), runtime=3(r_2), accessibility=y(r_3), bluechip=y(r_4)

id	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
sim(C,p_i)	**.66**	**.91**	**.92**	.41	**.88**	.36	**.48**	.08

Table 4: Calculated *sim*(C,i_j) for {i_1, …, i_8}

Personalized repairs [Felfernig'09]

Example *Financial services* [Felfernig'09]

id	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
sim(C,p_i)	**.66**	**.91**	**.92**	.41	**.88**	.36	**.48**	.08

Table 4: Calculated *sim*(C,i_j) for {i_1, …, i_8}

nearest neighbours

id	return-rate (p.a.)	run-time	risk level	shares percen-tage	accessi-bility	plow back earnings	blue chip
i_1	4.2	3.0	A	0	no	yes	yes
i_2	4.7	3.5	B	10	yes	no	yes
i_3	4.8	3.5	A	10	yes	yes	yes
i_5	4.3	3.5	A	0	yes	yes	yes
i_7	6.7	6.0	C	50	yes	no	no

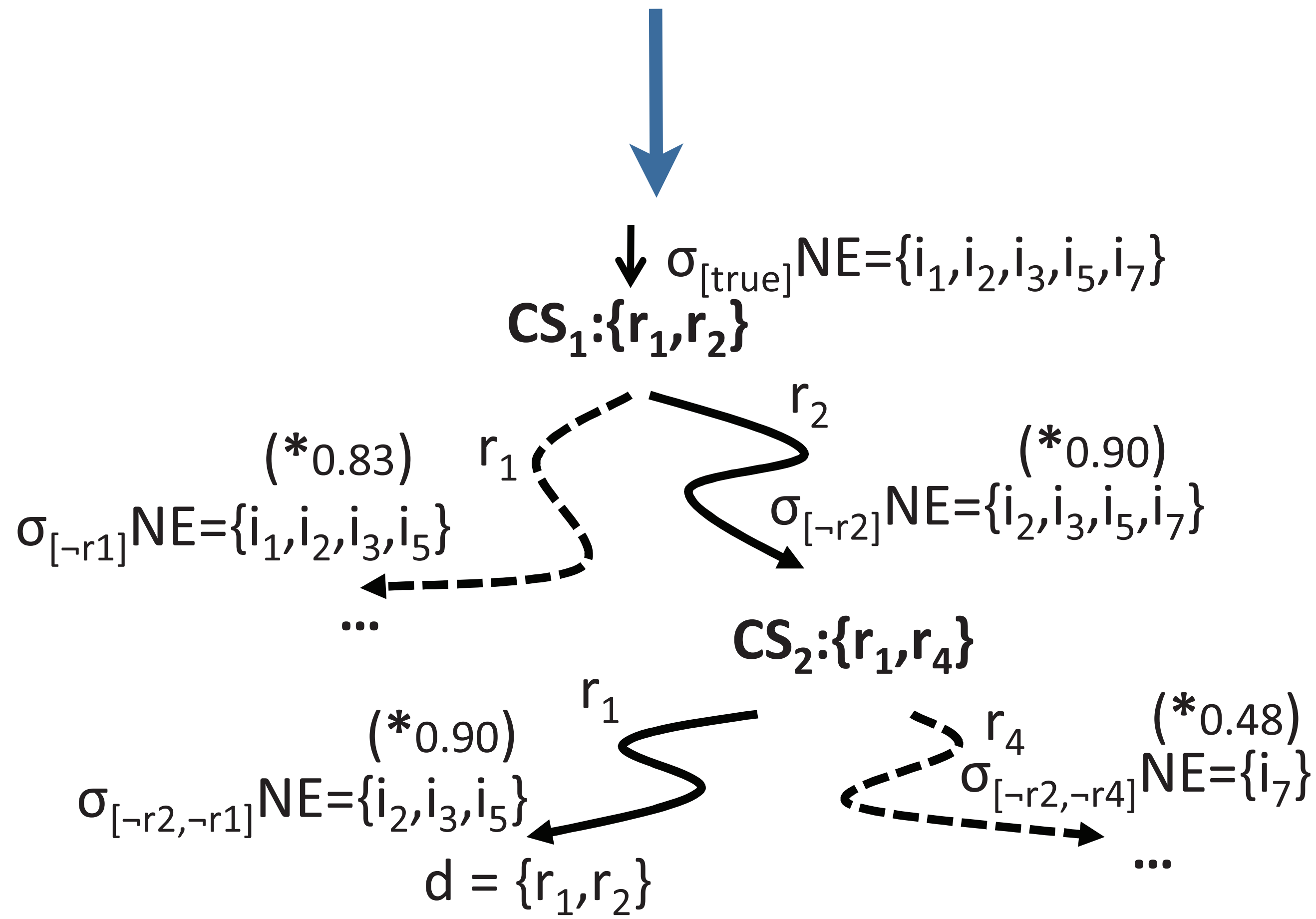
Table 3: Calculated nearest neighbors {i_1,i_2,i_3,i_5,i_7}

Personalized repairs [Felfernig'09]

Example *Financial services* [Felfernig'09]

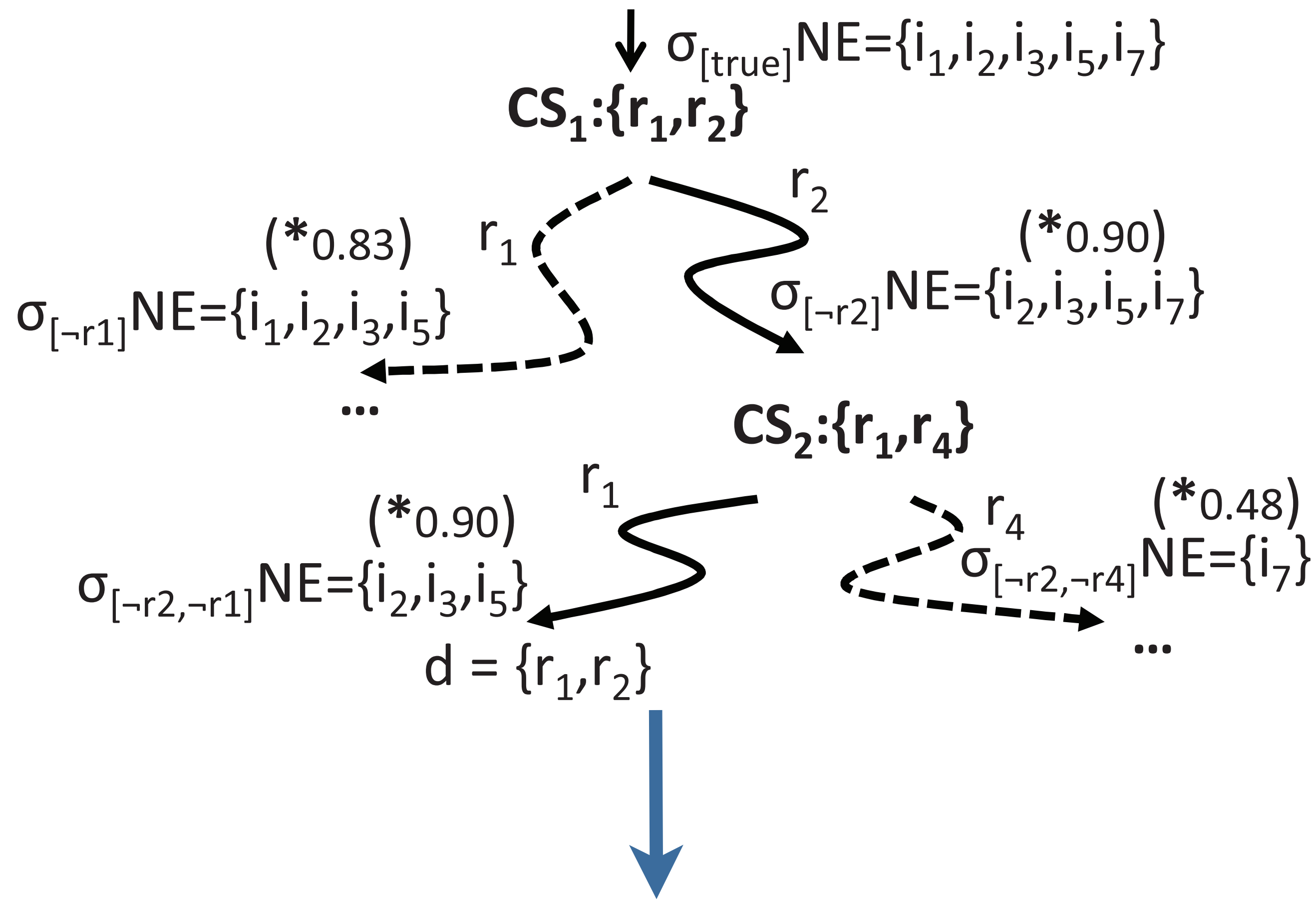
id	return-rate (p.a.)	run-time	risk level	shares percen-tage	accessi-bility	plow back earnings	blue chip
i_1	4.2	3.0	A	0	no	yes	yes
i_2	4.7	3.5	B	10	yes	no	yes
i_3	4.8	3.5	A	10	yes	yes	yes
i_5	4.3	3.5	A	0	yes	yes	yes
i_7	6.7	6.0	C	50	yes	no	no

Table 3: Calculated nearest neighbors $\{i_1,i_2,i_3,i_5,i_7\}$



Personalized repairs [Felfernig'09]

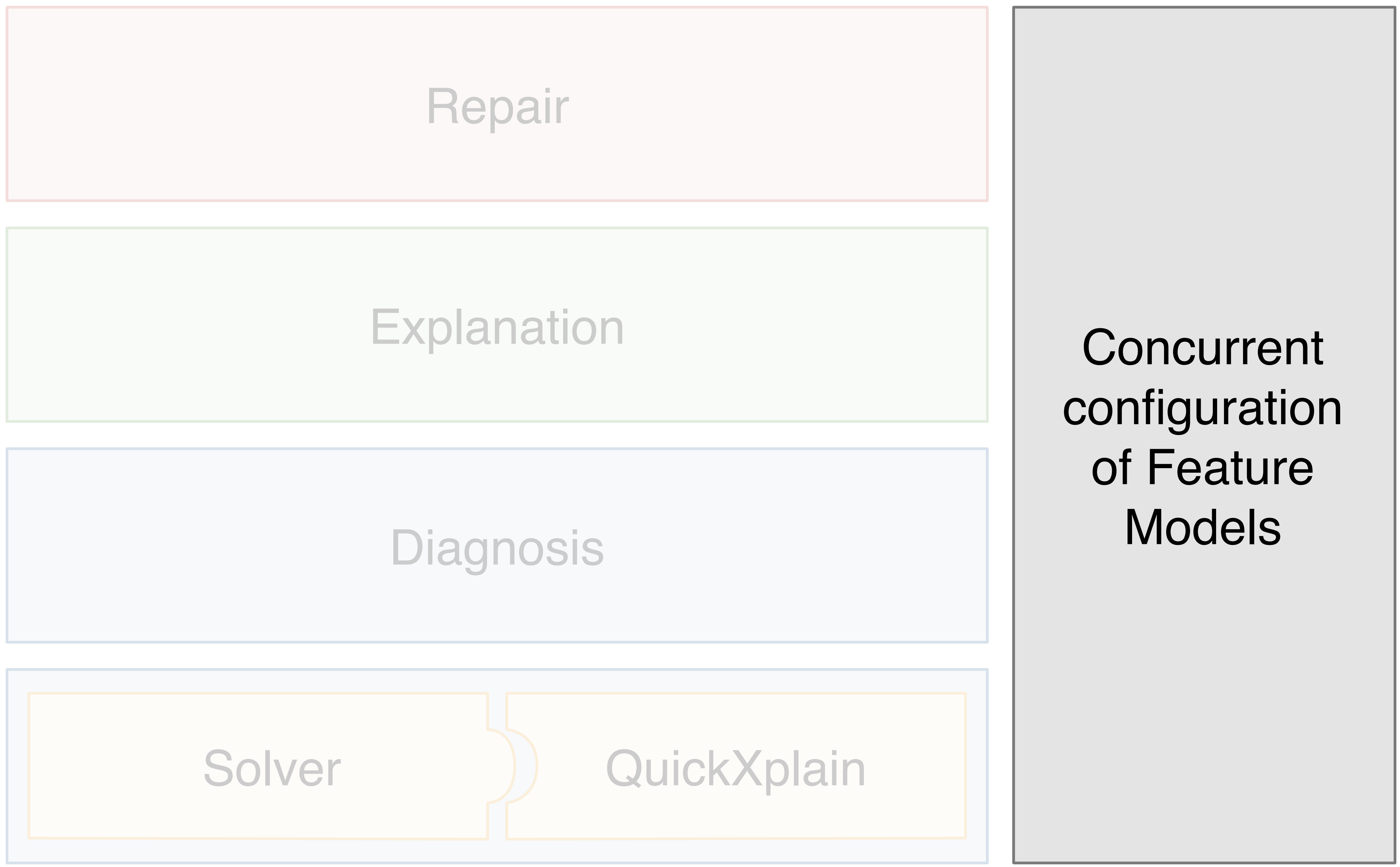
Example *Financial services* [Felfernig'09]



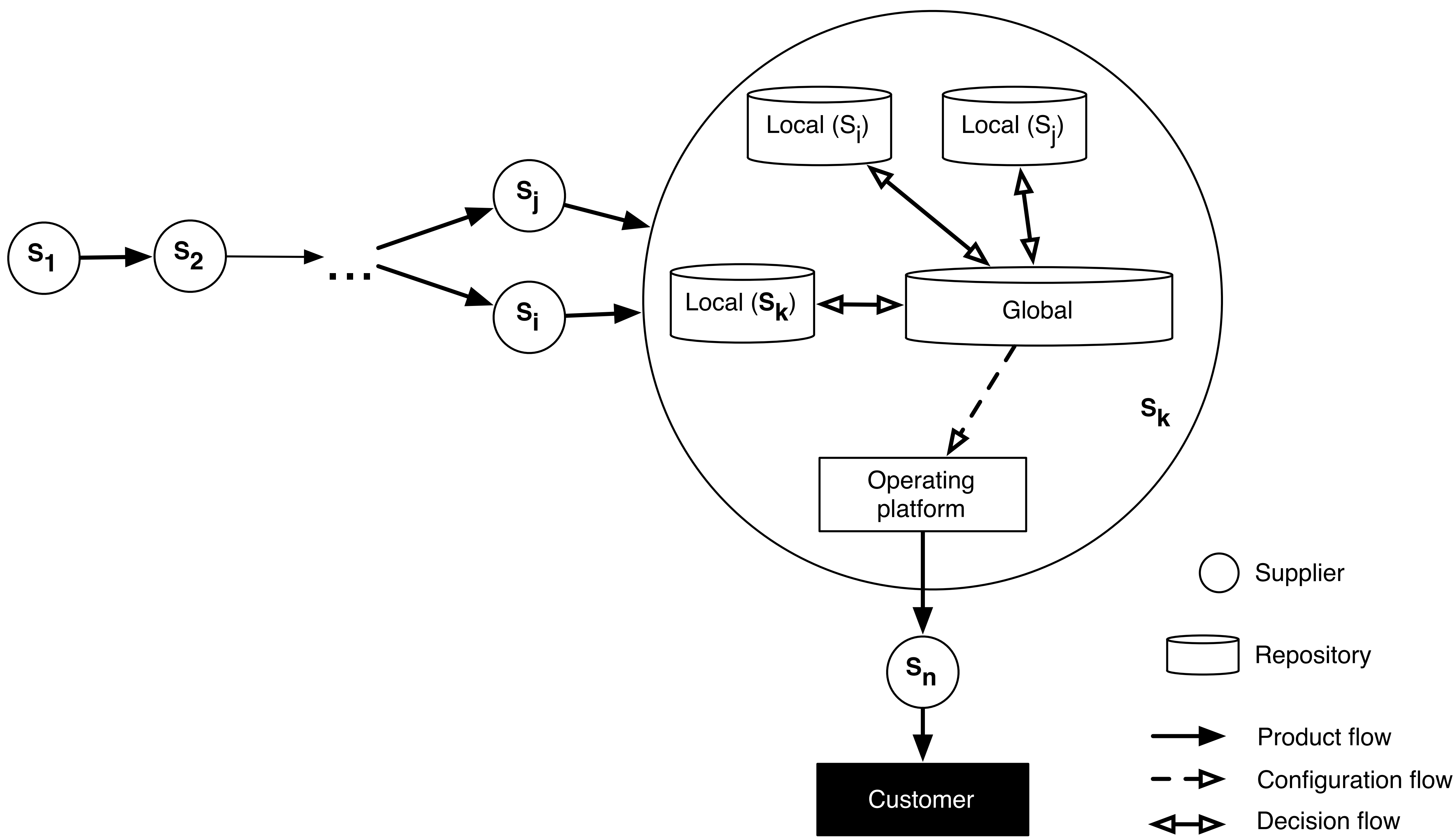
R = {accessibility=y(r_3), bluechip=y(r_4)}

F = {return-rate≥5.5(r_1), runtime=3(r_2)}

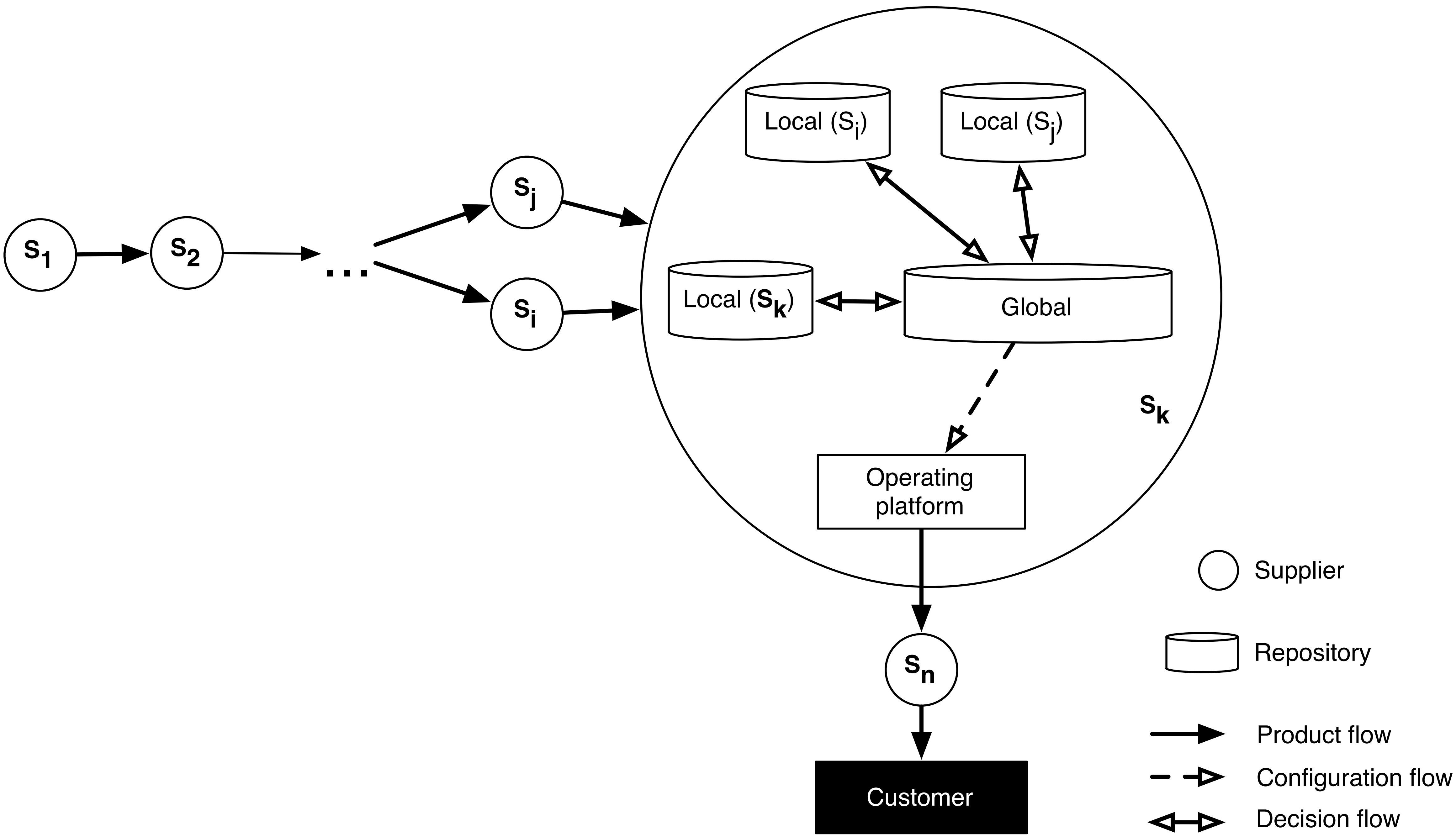
Repairs = {{return-rate=4.7, runtime=3.5}, {return-rate=4.8, runtime=3.5}, {return-rate=4.3, runtime=3.5}}


Repair
Explanation
Diagnosis
Solver
QuickXplain
Concurrent configuration of Feature Models

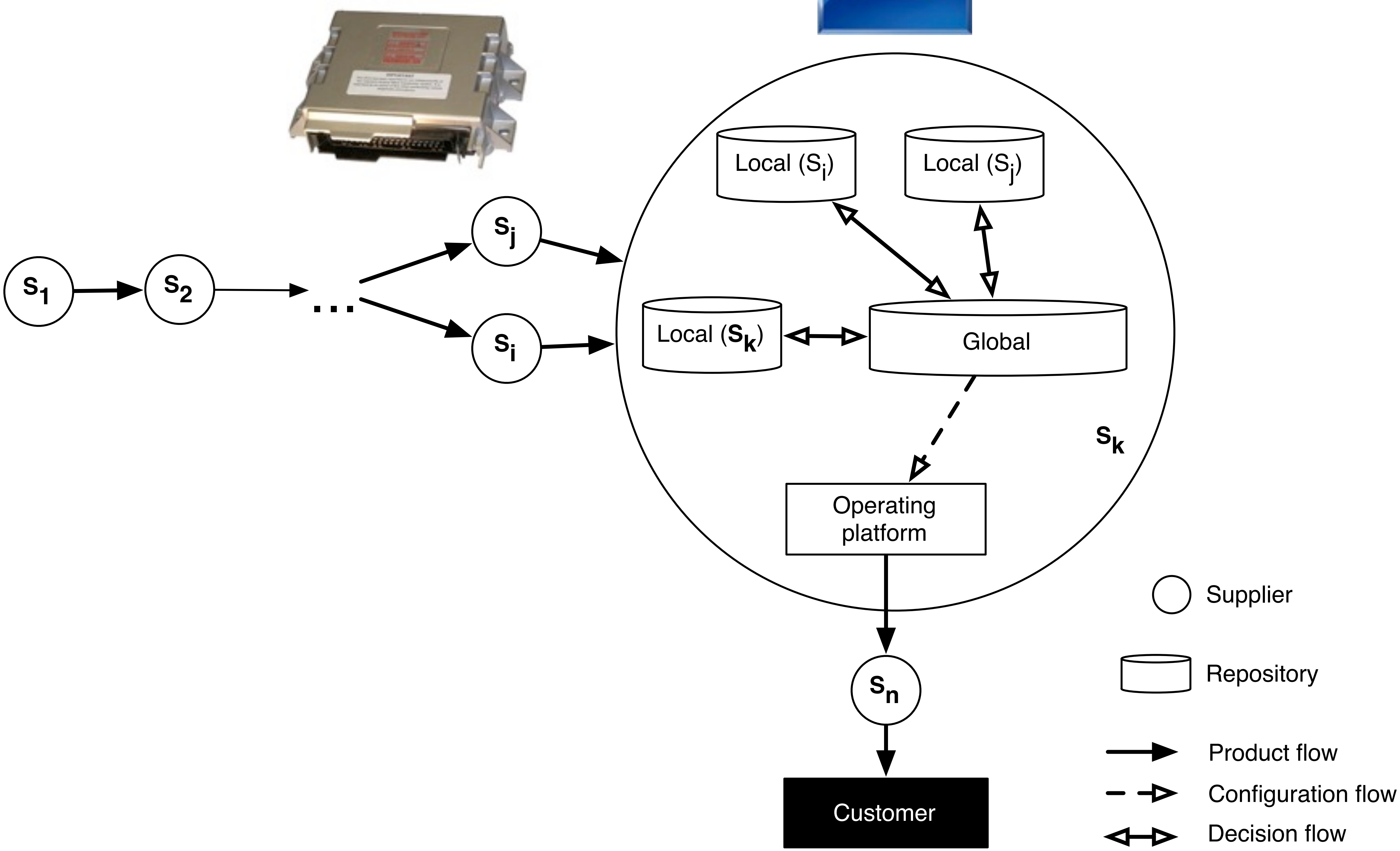
Concurrent configuration



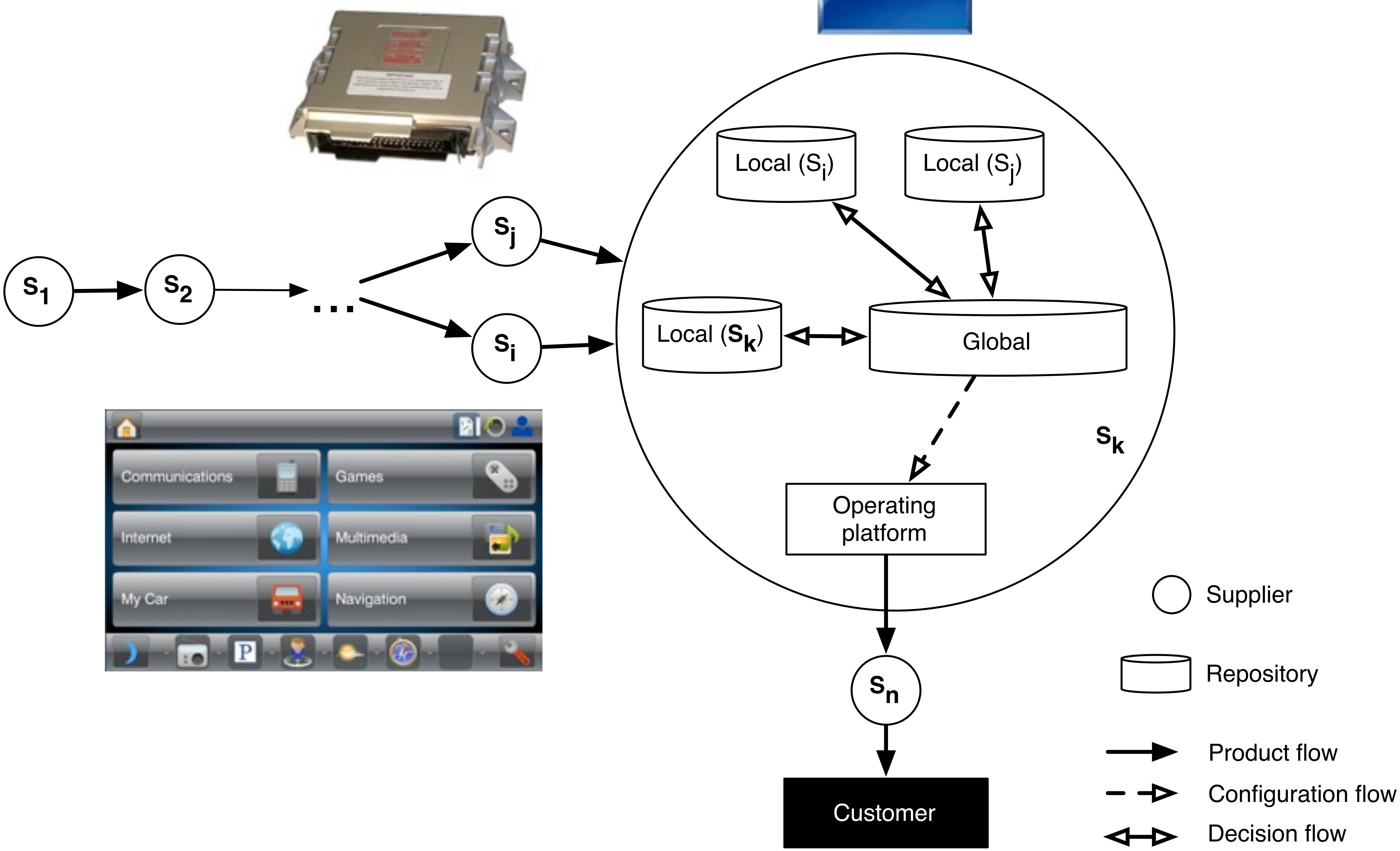
Concurrent configuration



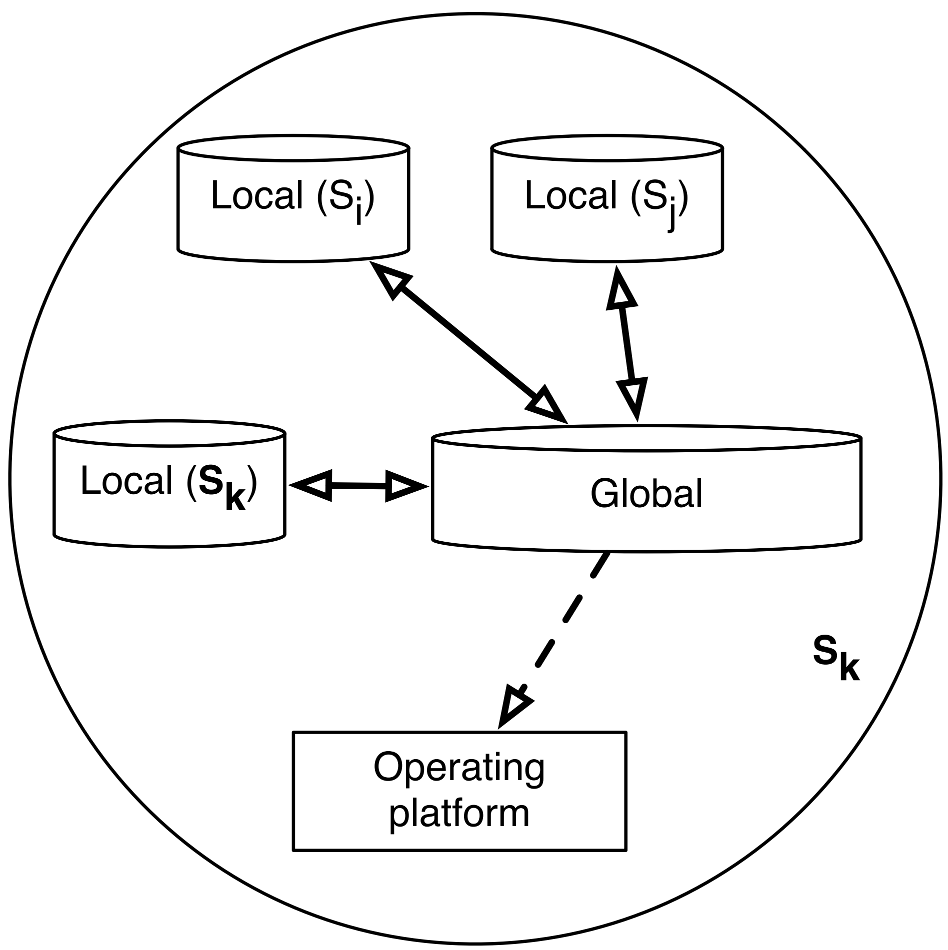
Concurrent configuration



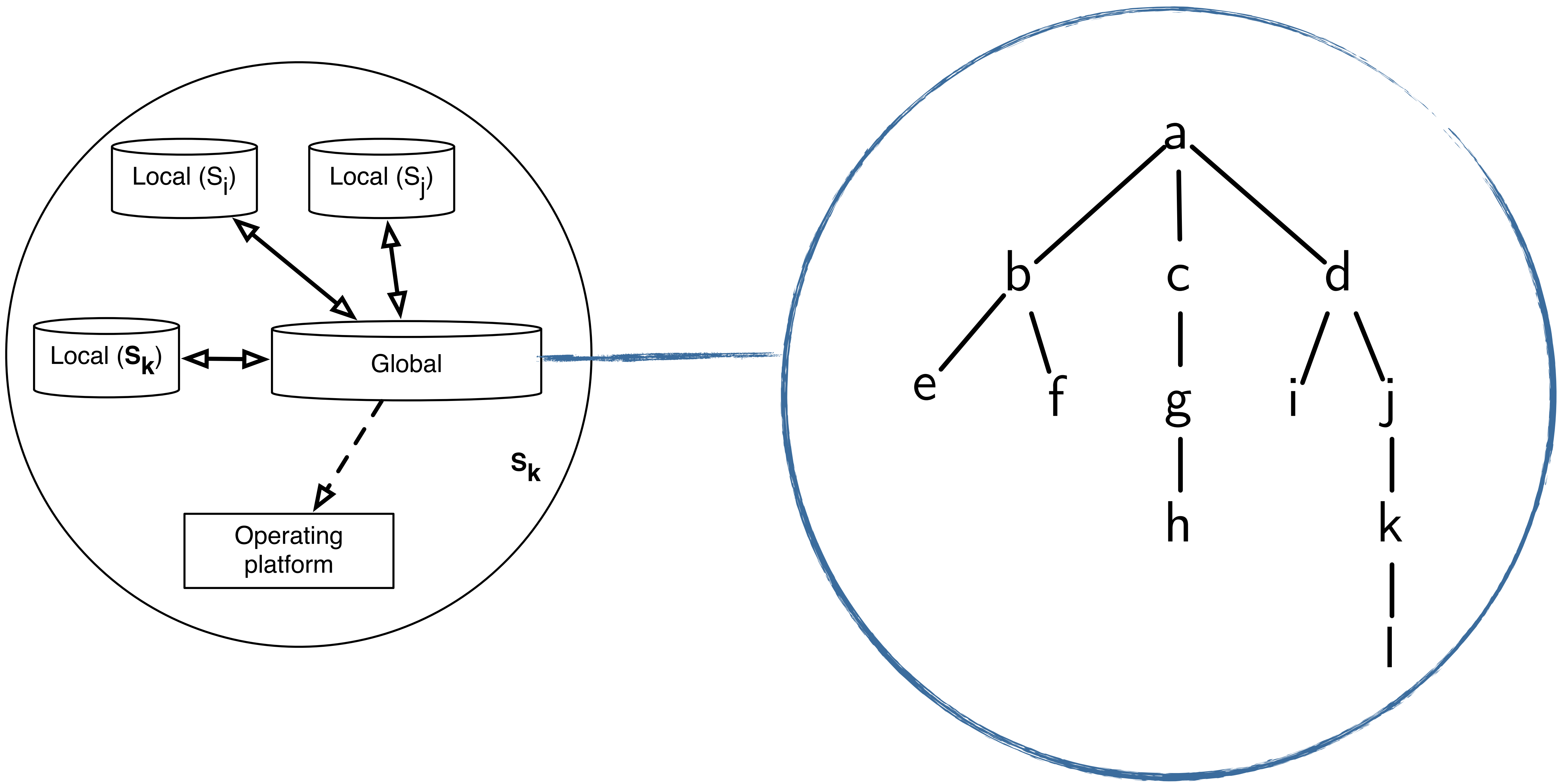
Concurrent configuration



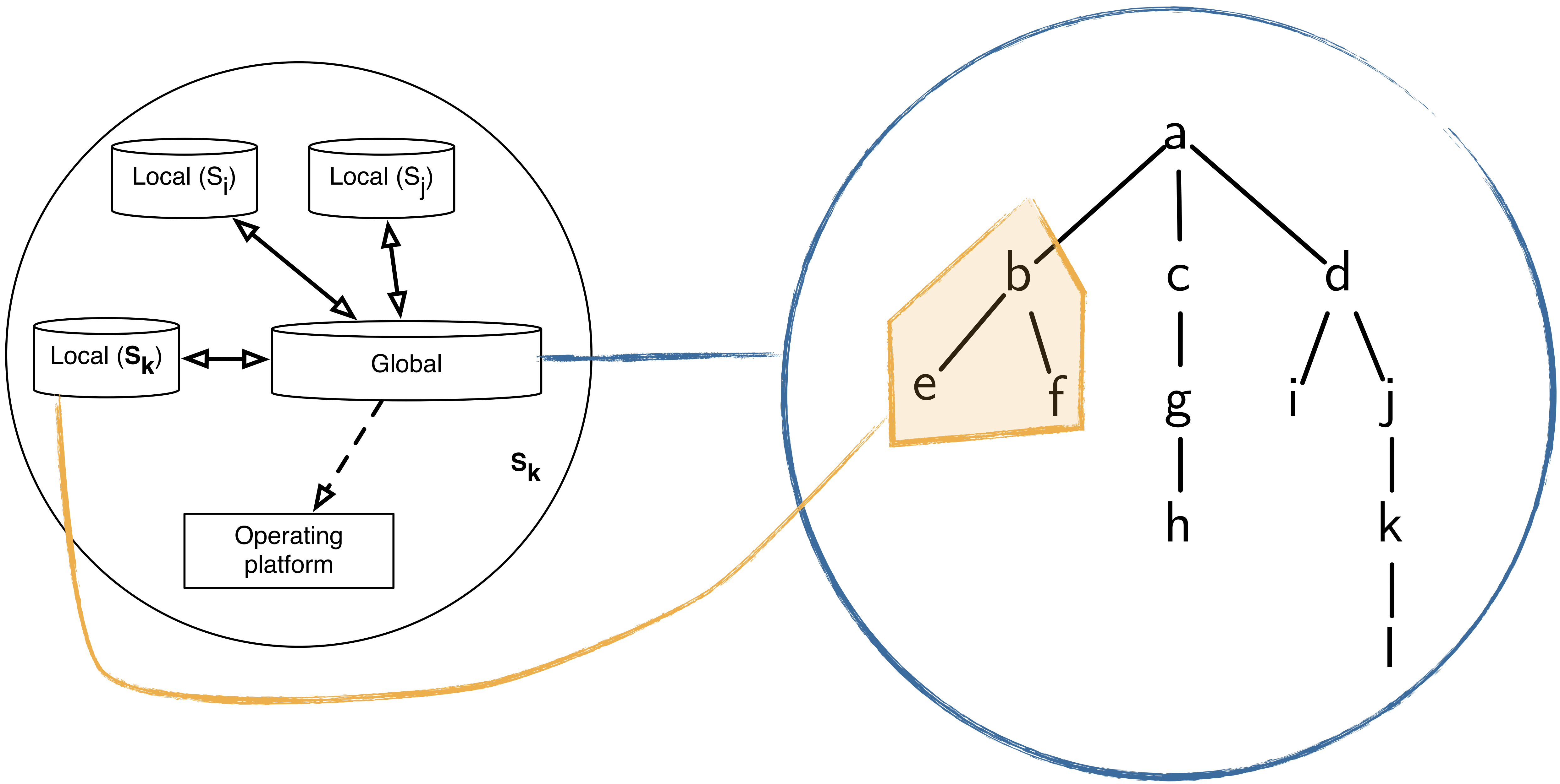
Concurrent configuration of FM



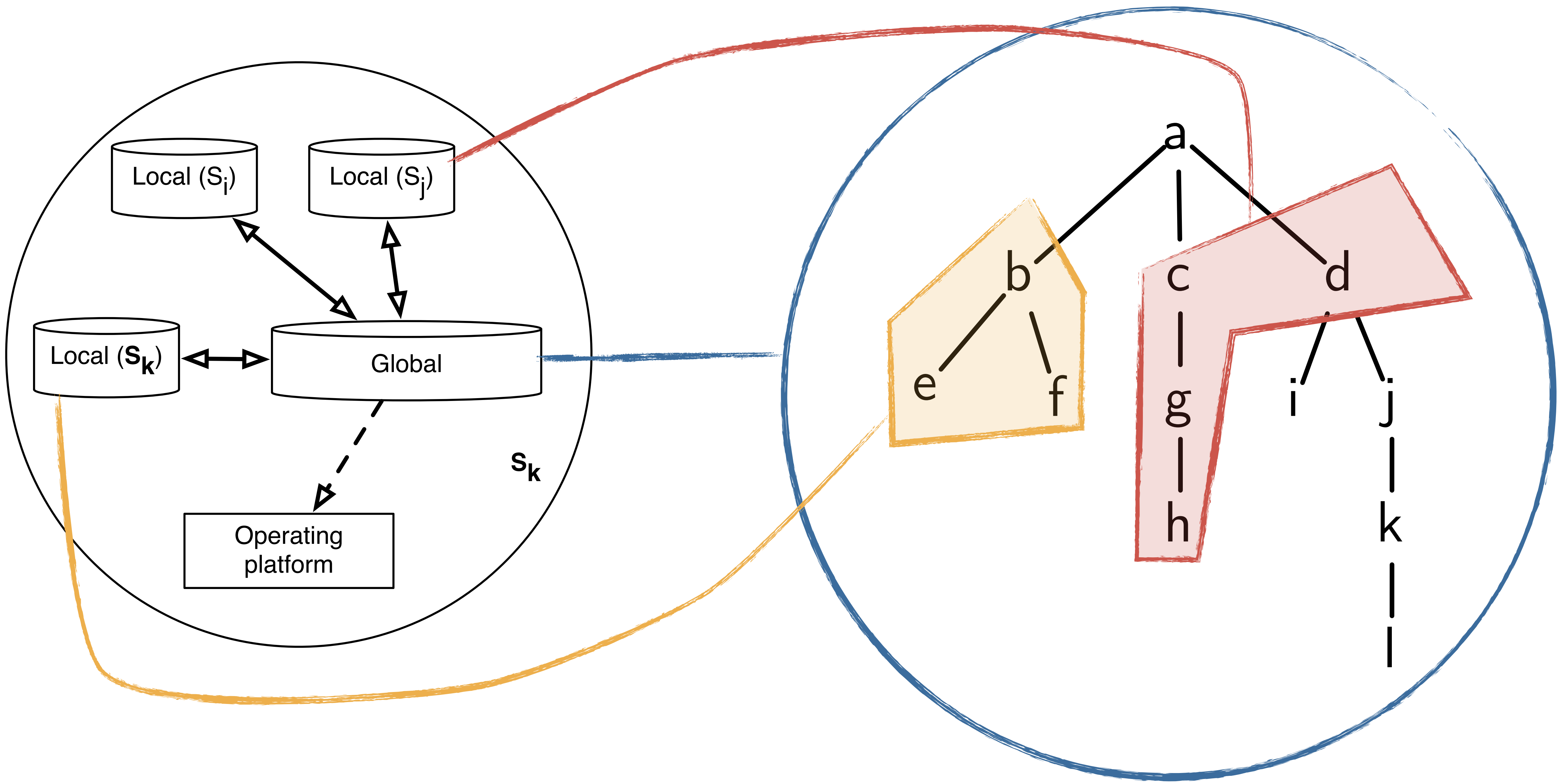
Concurrent configuration of FM



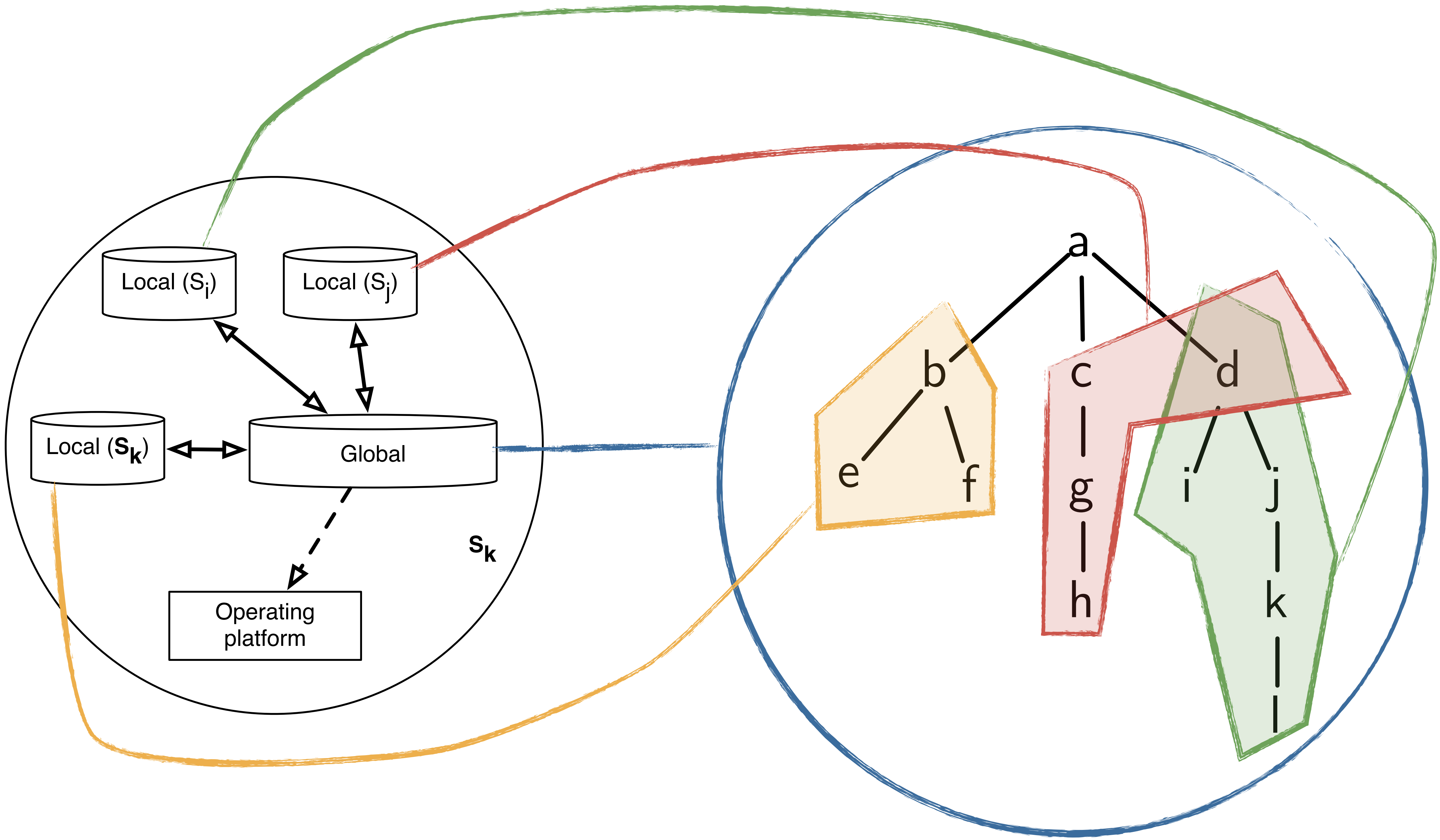
Concurrent configuration of FM



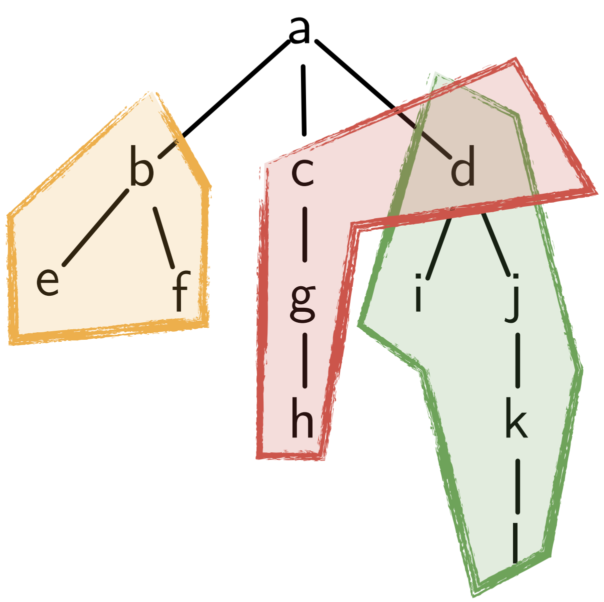
Concurrent configuration of FM



Concurrent configuration of FM

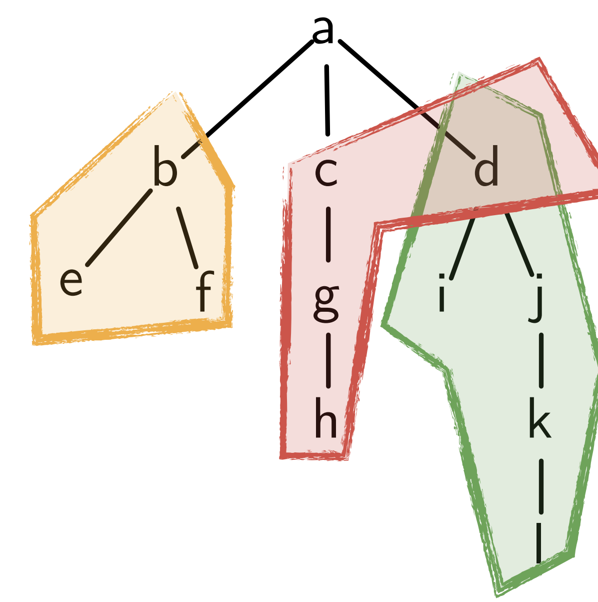


Concurrent configuration of FM

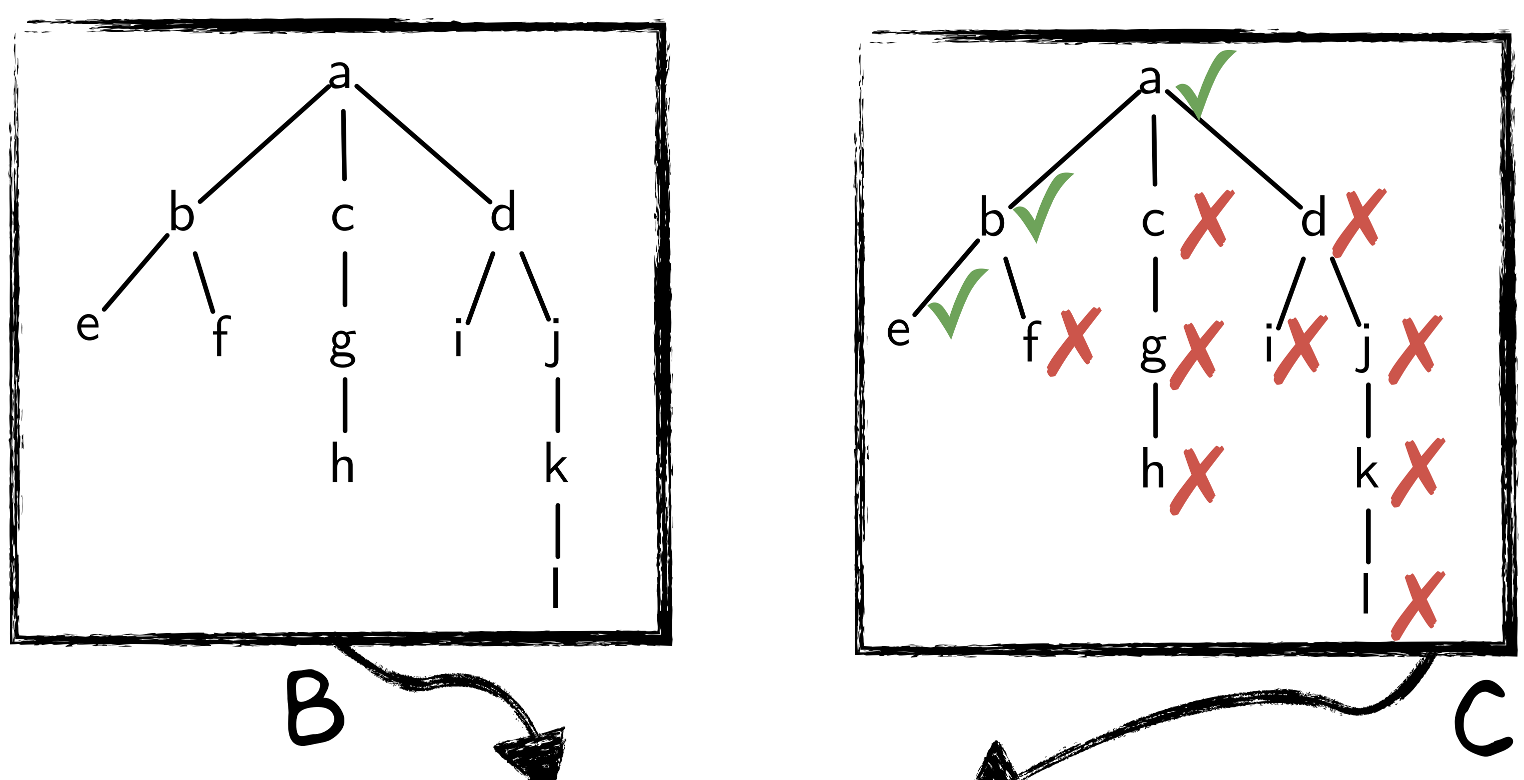


- Research questions:
  - How to detect constraint violations?
  - How to explain the cause of a conflict?

Concurrent configuration of FM

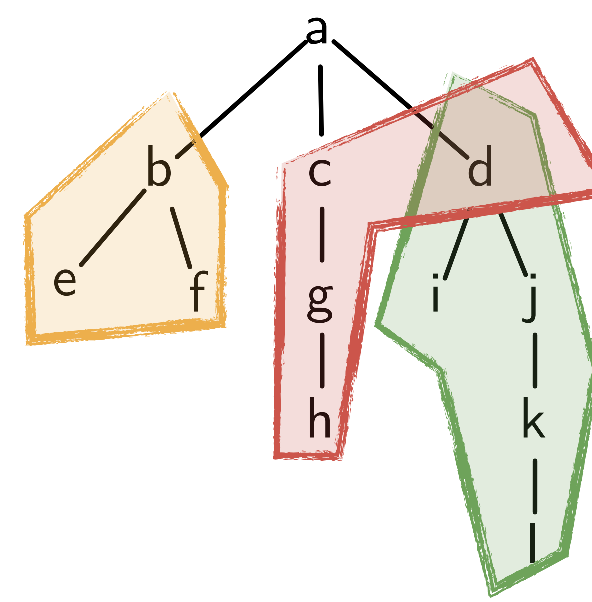


- Research questions:
  - How to detect constraint violations?
  - How to explain the cause of a conflict?

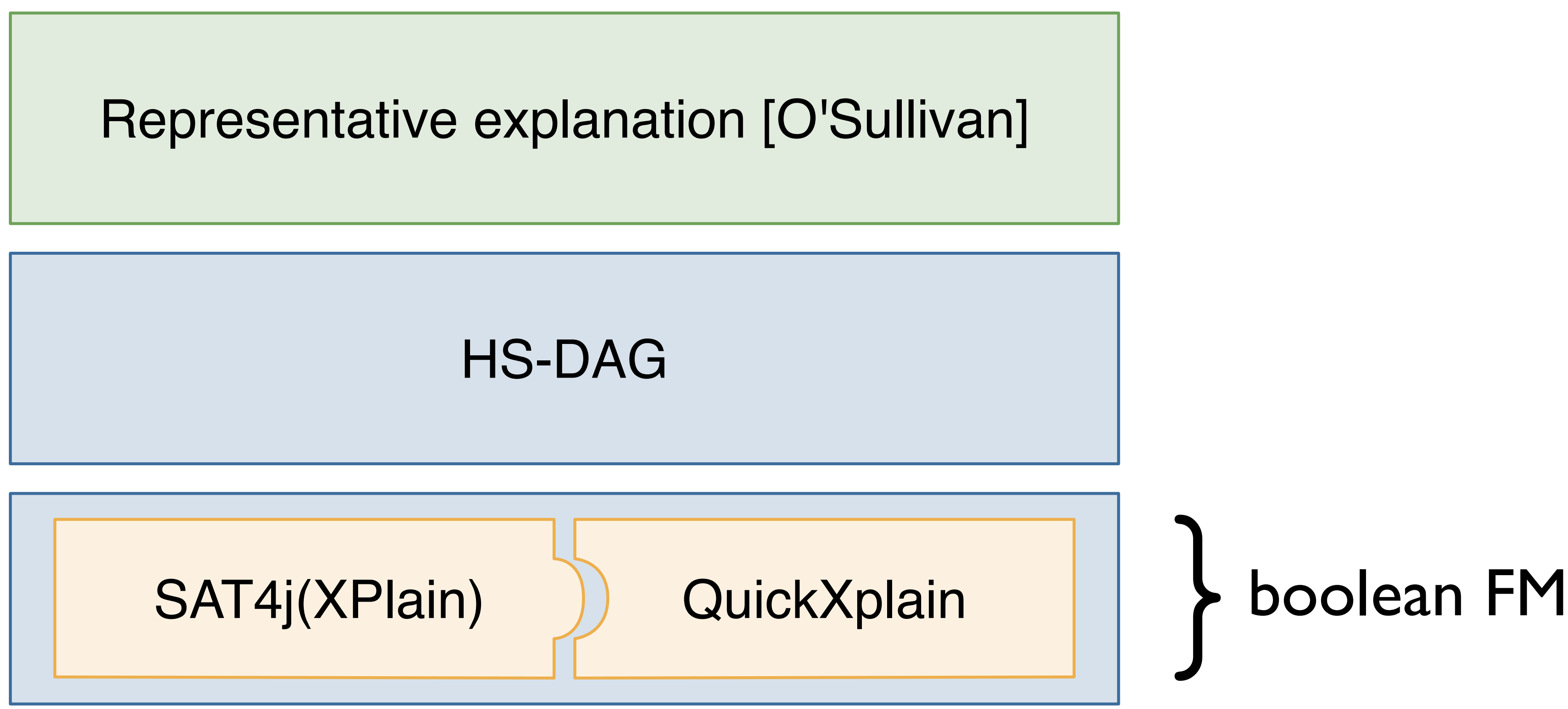


Configuration = $\Phi \wedge \{a, b, \neg c, \neg d, e \ldots\}$

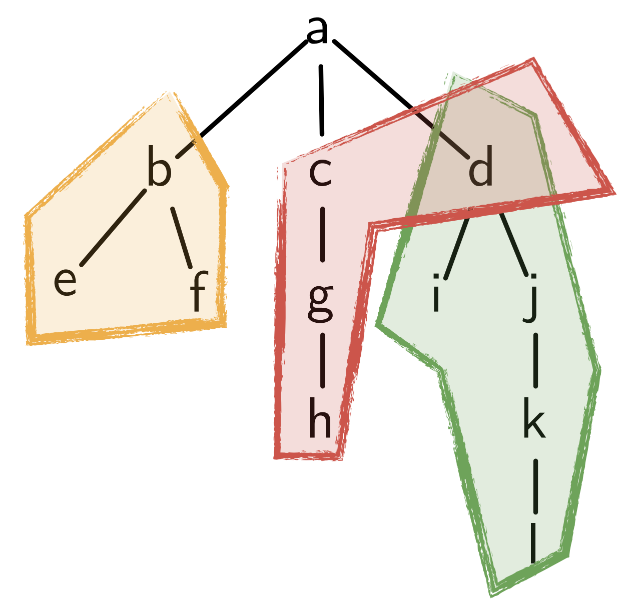
Concurrent configuration of FM



- Research questions:
  - How to detect constraint violations?
  - How to explain the cause of a conflict?



Concurrent configuration of FM



- Research questions:
  - How to detect constraint violations?
  - How to explain the cause of a conflict?

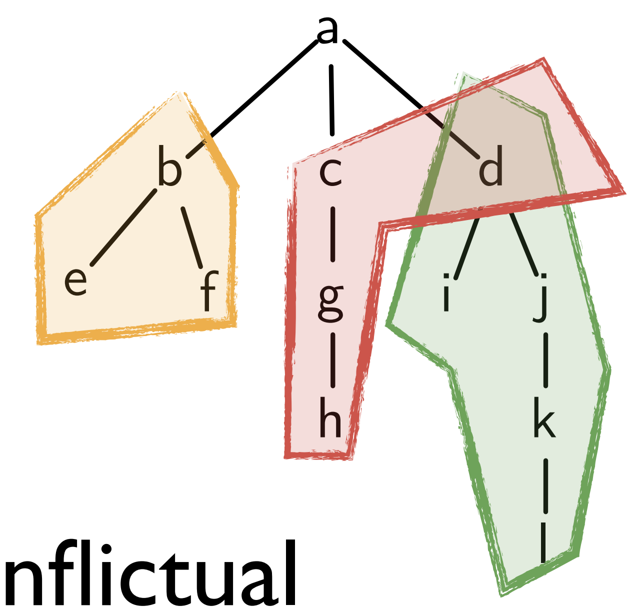
Representative explanation [O'Sullivan]

HS-DAG

SMT QuickXplain

numeric FM

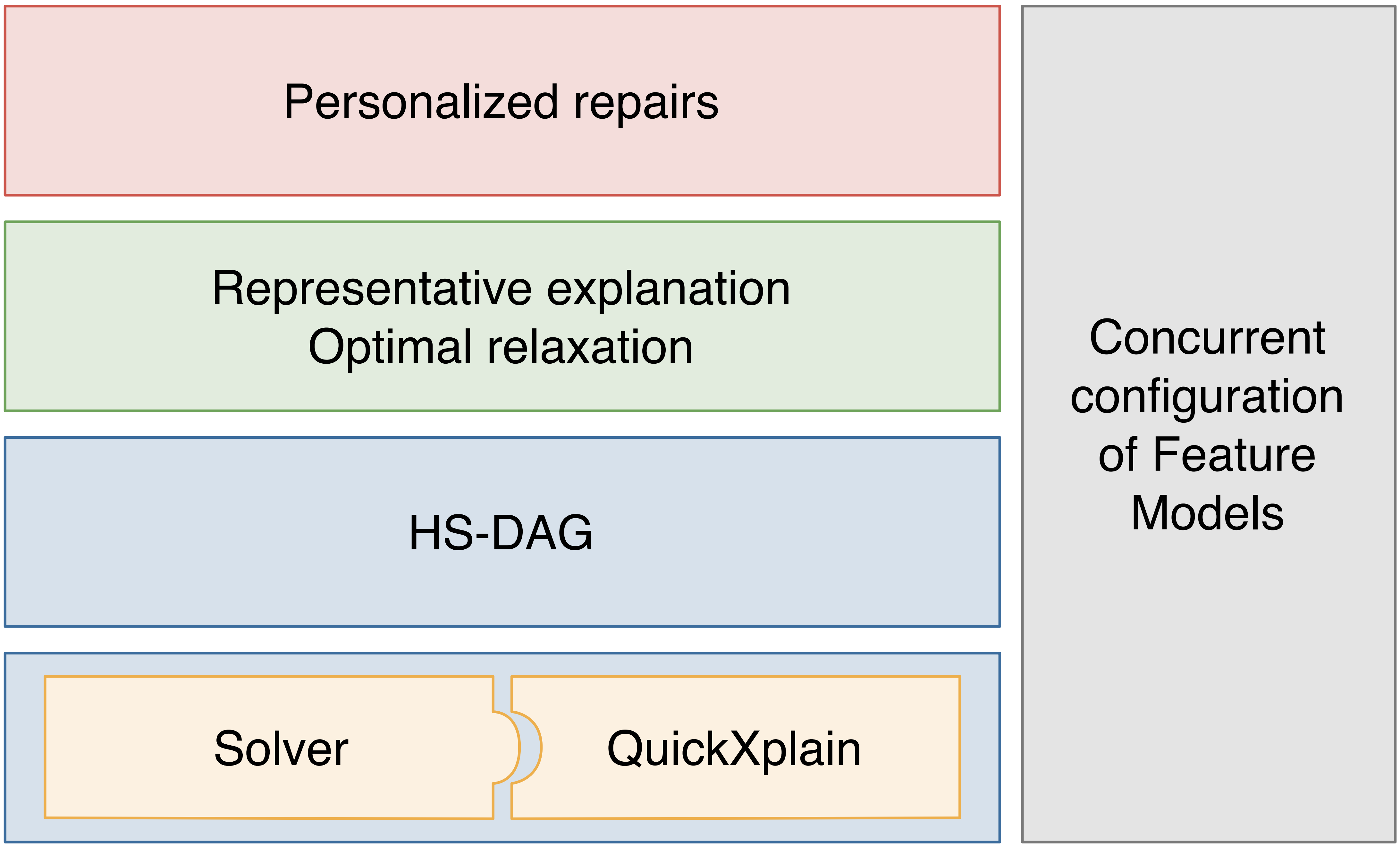
Concurrent configuration of FM



- Research questions:
  - How to differentiate manually and automatically resolvable decisions?
  - How to automate the resolution of conflictual decisions?

    Personalized repairs: complete product set?

  - How to perform n-way conflict resolution?
  - How to control the impact of conflict resolution?


Personalized repairs
Representative explanation
Optimal relaxation
HS-DAG
Solver
QuickXplain
Concurrent configuration of Feature Models