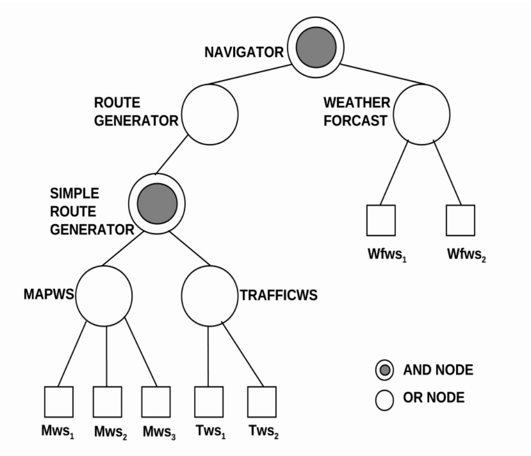
Helpful Behavior Based on Trust for Web Services

Ioan Alfred Letia, Radu Răzvan Slăvescu

Technical University of Cluj-Napoca
Department of Computer Science

Motivating Example

Recipe Composition



Outline

Outline

Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipe Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work

Outline

Introduction

Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost

Introduction

Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost

Introduction

Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost

Introduction

Issues when building composite WS

1. decompose the complex goal task into simpler ones, up to a basic level, when they can be solved by existing WSs
2. select a particular WS provider for every basic task

Criteria to be met

1. functional: producing the intended output
2. non-functional: quality aspects like response time, compliance or cost

Introduction

Suggested Approach

Developing a composite WS = solve, with maximum likelihood, a joint goal by a set of cooperative agents like WS architect and a consultant. Targets:

1. an abstract plan (solution)
2. and a concrete list of WS providers for every step of it

Types of decisions

- architect: choose each service and its provider
- consultant: adopt (or not) a helpful behavior (suggest an alternative to maximize the chances of success, while not providing its partner with more information than paid)

Introduction

Suggested Approach

Developing a composite WS = solve, with maximum likelihood, a joint goal by a set of cooperative agents like WS architect and a consultant. Targets:

1. an abstract plan (solution)
2. and a concrete list of WS providers for every step of it

Types of decisions

- architect: choose each service and its provider
- consultant: adopt (or not) a helpful behavior (suggest an alternative to maximize the chances of success, while not providing its partner with more information than paid)

Introduction

Contributions

1. employing a ”Belief Recipe Tree (BRT)” for making decisions
2. employing prior experiences-based trust, for the values required by the BRT
3. revisiting it after each experience

Sentences and opinions

Opinion

Proposition $x \rightarrow$ opinion $\omega_x =< b_x, d_x, u_x, a_x >$, (belief, disbelief, uncertainty, prior probability about the truth of x)

- binomial opinions $\rightarrow$ beta probability distribution function
- r evidence supporting x and s supporting $\neg x$ $\rightarrow$

$$\begin{cases} b = \frac{r}{r+s+2} \\ d = \frac{s}{r+s+2} \\ u = \frac{2}{r+s+2} \end{cases}$$

- probability expectation value:

$$E(\omega_x) = b_x + a_x u_x$$

Sentences and opinions

Opinion

Proposition $x \rightarrow$ opinion $\omega_x =< b_x, d_x, u_x, a_x >$, (belief, disbelief, uncertainty, prior probability about the truth of x)

- binomial opinions $\rightarrow$ beta probability distribution function
- r evidence supporting x and s supporting $\neg x$ $\rightarrow$

$$\begin{cases} b = \frac{r}{r+s+2} \\ d = \frac{s}{r+s+2} \\ u = \frac{2}{r+s+2} \end{cases}$$

- probability expectation value:

$$E(\omega_x) = b_x + a_x u_x$$

Sentences and opinions

Opinion

Proposition $x \rightarrow$ opinion $\omega_x =< b_x, d_x, u_x, a_x >$, (belief, disbelief, uncertainty, prior probability about the truth of x)

- binomial opinions $\rightarrow$ beta probability distribution function
- r evidence supporting x and s supporting $\neg x \rightarrow$

$$\begin{cases} b = \frac{r}{r+s+2} \\ d = \frac{s}{r+s+2} \\ u = \frac{2}{r+s+2} \end{cases}$$

- probability expectation value:

$$E(\omega_x) = b_x + a_x u_x$$

Sentences and opinions

Opinion

Proposition $x \rightarrow$ opinion $\omega_x =< b_x, d_x, u_x, a_x >$, (belief, disbelief, uncertainty, prior probability about the truth of x)

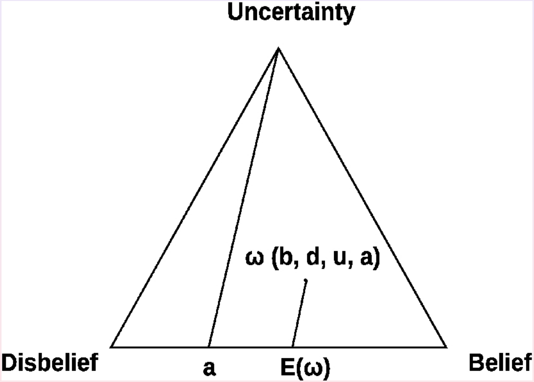
- binomial opinions $\rightarrow$ beta probability distribution function
- r evidence supporting x and s supporting $\neg x$ $\rightarrow$

$$\begin{cases} b = \frac{r}{r+s+2} \\ d = \frac{s}{r+s+2} \\ u = \frac{2}{r+s+2} \end{cases}$$

- probability expectation value:

$$E(\omega_x) = b_x + a_x u_x$$

Sentences and opinions



Sentences and opinions

Operators

1. conjunction $\wedge$: given $\omega_x = (b_x, d_x, u_x, a_x)$ and $\omega_y = (b_y, d_y, u_y, a_y)$, their conjunction $\omega_{x \wedge y}$, written $\omega_x \cap \omega_y$, has the following components:

$$\begin{cases} b_{x\wedge y} = b_x b_y + \frac{(1-a_x)a_y b_x u_y + a_x(1-a_y)u_x b_y}{1-a_x a_y} \\ d_{x\wedge y} = d_x + d_y - d_x d_y \\ u_{x\wedge y} = u_x u_y + \frac{(1-a_y)b_x u_y + (1-a_x)u_x b_y}{1-a_x a_y} \\ a_{x\wedge y} = a_x a_y \end{cases}$$

2. disjunction $\vee$: similar

BRT Structure

Purpose

- defines a probability distribution over possible recipes for completing the action in the tree's root
- each node in a BRT has an action attached
- allows agents to reason *efficiently* about the possible recipes for performing an action: $\mathcal{O}(nm)^d$, (n = number of potential recipes for an action, m = average number of steps per action, d = number of levels of decomposition)
- deals with opinions rather than mere probabilities as its Probabilistic counterpart

Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipe Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work

Structure

Node types

- terminal nodes: atomic tasks
- non-terminal nodes: composed tasks

Node information

- an opinion describing the trust level the evaluator has in the success of the task represented by that node
- a cost, which estimates how much the plan effector must pay in order to have the task corresponding to the node completed
- an income modeling the benefit obtained if the task is accomplished.

Structure

Node types

- terminal nodes: atomic tasks
- non-terminal nodes: composed tasks

Node information

- an opinion describing the trust level the evaluator has in the success of the task represented by that node
- a cost, which estimates how much the plan effector must pay in order to have the task corresponding to the node completed
- an income modeling the benefit obtained if the task is accomplished.

BRT structure

Structure

1. leaf nodes: atomic tasks (selecting a specific atomic web service)
2. (intermediate) AND: all its children must be completed (multi-step recipe for a composed web service)
3. (intermediate) OR: alternatives for achieving a specific goal (making a nondeterministic choice of one alternative over the other in case one need a web service of type T and two concrete web services of that type, Tws_1 and Tws_2 are available but just one needs to be chosen)

Outline

BRT structure

Functions attached

1. context: all beliefs an agent bases its decision on at moment T when action α is done.
2. function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent G_1 can bring about the atomic action β within context C_β.
3. function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent G_1 when the basic level action β is done by agent G_2 within context C_β.
4. function $V(G_1, \alpha, C_\alpha)$ gives the utility form G_1 if action α is performed within context C_α;

BRT structure

Functions attached

1. context: all beliefs an agent bases its decision on at moment T when action α is done.
2. function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent G_1 can bring about the atomic action β within context C_β.
3. function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent G_1 when the basic level action β is done by agent G_2 within context C_β.
4. function $V(G_1, \alpha, C_\alpha)$ gives the utility form G_1 if action α is performed within context C_α;

BRT structure

Functions attached

1. context: all beliefs an agent bases its decision on at moment T when action α is done.
2. function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent G_1 can bring about the atomic action β within context C_β.
3. function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent G_1 when the basic level action β is done by agent G_2 within context C_β.
4. function $V(G_1, \alpha, C_\alpha)$ gives the utility form G_1 if action α is performed within context C_α;

BRT structure

Functions attached

1. context: all beliefs an agent bases its decision on at moment T when action α is done.
2. function $cba.basic(G_1, \beta, C_\beta)$ gives the probability that agent G_1 can bring about the atomic action β within context C_β.
3. function $cba.cost(G_1, G_2, \beta, C_\beta)$ returns the cost for agent G_1 when the basic level action β is done by agent G_2 within context C_β.
4. function $V(G_1, \alpha, C_\alpha)$ gives the utility form G_1 if action α is performed within context C_α;

BRT structure

Functions attached

1. Function $p\_CBA(BRT_{\alpha}, C_{\alpha})$ computes the probability of action α to succeed and returns *cba.basic* for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.
2. $Cost(G_i, BRT_{\alpha}, C_{\alpha})$ computes the expected cost payed by agent G_i and returns *cost.basic* for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.
3. $Eval(BRT_{\alpha})$ computes expected utility based on costs, incomes and probabilities

BRT structure

Functions attached

1. Function $p\_CBA(BRT_\alpha, C_\alpha)$ computes the probability of action α to succeed and returns *cba.basic* for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.
2. $Cost(G_i, BRT_\alpha, C_\alpha)$ computes the expected cost payed by agent G_i and returns *cost.basic* for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.
3. $Eval(BRT_\alpha)$ computes expected utility based on costs, incomes and probabilities

BRT structure

Functions attached

1. Function $p\_CBA(BRT_\alpha, C_\alpha)$ computes the probability of action α to succeed and returns *cba.basic* for leaf nodes, a product of children probabilities for AND nodes and a weighted average of children probabilities for an OR node.
2. $Cost(G_i, BRT_\alpha, C_\alpha)$ computes the expected cost payed by agent G_i and returns *cost.basic* for leaf nodes, the sum of children costs for AND nodes and a weighted average of children costs for an OR node.
3. $Eval(BRT_\alpha)$ computes expected utility based on costs, incomes and probabilities

Outline

BRT structure

Commitment

Agent G_1 is committed to α iff G_1 believes that BRT_α maximizes group utility:

$$\exists BRT_\alpha\ BEL(G_1, \forall BRT_\beta\ BRT_\beta \neq BRT_\alpha \Rightarrow Eval(BRT_\beta) \leq Eval(BRT_\alpha)) \wedge Int.Th(G_1, SelectedBRT(BRT_\alpha))$$

BRT structure

Helpful behavior

If agent is committed to α and believes that sending information o to its partner will increase group utility, it will do so:

if $Committed(G_1, GR, \alpha)$ **then**

$BRT_\alpha = PredictBRT(G_1, GR, \alpha, C_{GR})$

$C_\beta = ContextUpdate(C_\beta, o)$

$BRT_\beta = PredictBRT(G_1, G_2, \beta, C_\beta)$

$BRT^o_\alpha = BRTReplace(BRT_\alpha, BRT_\beta)$

$utility = Eval(BRT^o_\alpha) - Eval(BRT_\alpha)$

endif

if $utility \geq CommunicationCost(G_2)$ **then**

$Int.To(G_1, Communicate(G_1, G_2, o))$

endif

Outline

1. Problem statement
2. Subjective Logic Based Trust
3. Belief Receipe Tree
   - Node types
   - Functions
   - Commitment
   - BRT example
4. Conclusions and Future Work

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

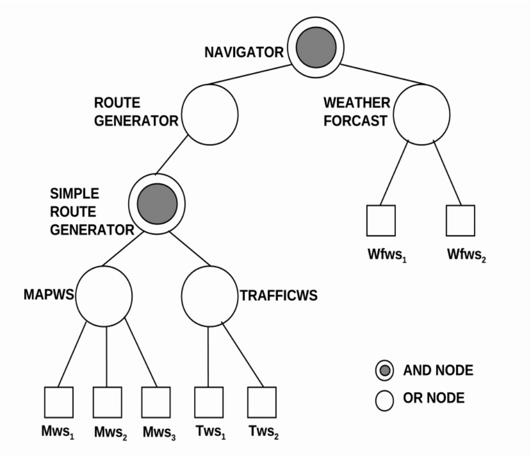
An example: Route generator design

1. an architect *A* and a consultant *C* want to build a *Navigator* WS (which generates car routes for drivers)
2. info for route building: paths + traffic + weather forcast
3. WS offering info: Mws_1, Mws_2, Mws_3 (map WS), Tws_1, Tws_2 (traffic info WS), $Wfws_1$, $Wfws_2$ (weather forcast).
4. system = *Route Generator* + *Weather Forcast*.
5. *Route Generator* = *Mapws* (map info) + *Trafficws* (traffic info);
6. *Weather Forcast* = weather info
7. the customer agreed to pay $30K for the *Route Generator*, $2K for the *Weather Forcast* and separately $40K for the final delivery

An example: Route generator design

Question: if consultant *C* has just learned that WS *Complex Route Generator* does the jobs of *Mapws* and *Trafficws* in one step andarchitect *A* is not aware of this, should *C* convey this new piece of knowledge to *A* or not?

No complex route generator



Atomic WS evidence and opinions

WS	r	s	b	d	u	a
Mws_1	1	1	0.250	0.250	0.500	0.500
Mws_2	3	1	0.500	0.167	0.333	0.500
Mws_3	1	2	0.200	0.400	0.400	0.500
Tws_1	5	1	0.625	0.125	0.250	0.500
Tws_2	3	2	0.428	0.286	0.286	0.500
$Wfws_1$	78	22	0.765	0.216	0.019	0.500
$Wfws_2$	67	33	0.657	0.222	0.111	0.500

Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator) = \$5,202.40$
5. if *Complex Route Generator* costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is $7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator)$ = \$5,202.40
5. if *Complex Route Generator* costs \$10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is \$7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator) = \$5{,}202.40$
5. if *Complex Route Generator* costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is $7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator)$ = \$5,202.40
5. if *Complex Route Generator* costs \$10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is \$7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

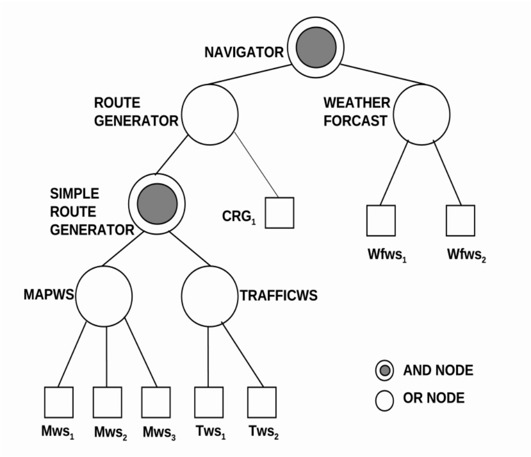
Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator)$ = \$5,202.40
5. if *Complex Route Generator* costs \$10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is \$7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

Deciding to cooperate

1. if, for *Mapws*, the descending branch opinions Mws_1, Mws_2, Mws_3 are $(0.2, 0.4, 0.4, 0.5)$, $(0.25, 0.25, 0.5, 0.5)$ and $(0.4, 0.2, 0.4, 0.5)$, then the *Mapws* opinion is $(0.433, 0.288, 0.279, 0.578)$.
2. *Simplerg*: computation gives $(0.345, 0.465, 0.19, 0.253)$.
3. *Navigator*: $(0.227, 0.635, 0.138, 0.553)$.
4. $Eval(Navigator)$ = $5,202.40
5. if *Complex Route Generator* costs $10,000 and has opinions of $(0.666, 0.167, 0.167, 0.500)$ and $(0.333, 0.0, 0.667, 0.500)$, the new value for $Eval(Navigator)$ is $7,919.38
6. Conclusion: it worth telling *A* about this, as long as the cost of communication is lower than the additional benefit.

With complex route generator



Deciding not to cooperate

1. if CRG_1 has failed twice, after adjusting the beliefs, we get the new values for $Eval(Navigator)$: \$5,202.40 and \$4,180.70 respectively.
2. Conclusion: in this case, C should not suggest the alternative

Observation

This is not feasible in the classical approach when the values of probabilities remain the same, thus the flexibility of this new approach

Deciding not to cooperate

1. if CRG_1 has failed twice, after adjusting the beliefs, we get the new values for $Eval(Navigator)$: \$5,202.40 and \$4,180.70 respectively.
2. Conclusion: in this case, C should not suggest the alternative

Observation

This is not feasible in the classical approach when the values of probabilities remain the same, thus the flexibility of this new approach

BRT versus PRT

In the same scenario, let us consider 2 situations:

1. $WFws_1$ has performance record $r = 300$, $s = 1000$
2. $WFws_1$ has performance record $r = 3$, $s = 1$

If a maximum threshold of uncertainty of 0.01 is considered:

1. situation 1 ($u = 0.001$): commit to weather forcast WS
2. situation 2 ($u = 0.333$): ask for more evidence

Observation

The classical approach gives the same solution since the same ratio r/s. This also illustrates the flexibility of this new approach

BRT versus PRT

In the same scenario, let us consider 2 situations:

1. $WFws_1$ has performance record $r = 300, s = 1000$
2. $WFws_1$ has performance record $r = 3, s = 1$

If a maximum threshold of uncertainty of 0.01 is considered:

1. situation 1 ($u = 0.001$): commit to weather forcast WS
2. situation 2 ($u = 0.333$): ask for more evidence

Observation

The classical approach gives the same solution since the same ratio r/s. This also illustrates the flexibility of this new approach

Conclusions

1. BRT extended with trust in form of subjective opinions to accomodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.

Conclusions

1. BRT extended with trust in form of subjective opinions to accomodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.

Conclusions

1. BRT extended with trust in form of subjective opinions to accomodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.

Conclusions

1. BRT extended with trust in form of subjective opinions to accomodate uncertainty of probability estimations (e.g. if just a small number of interactions have been observed)
2. this allows agents to decide whether to pay for more information or simply to rely on the available one
3. it also allows agents to adjust beliefs as further evidence becomes available
4. it offers a more realistic approach for the uncertainty present in designing complex SOA, as well as a better flexibility in decision making.

Future work

1. extensive experiments with this formalism
2. extend the formalism to multinomial (complex. mutually exclusive) opinions like ”The level of success is *poor/average/good/excellent*” as opposite to a mere *failure/success*
3. estimate base rates by learning class stereotypes

Future work

1. extensive experiments with this formalism
2. extend the formalism to multinomial (complex. mutually exclusive) opinions like ”The level of success is *poor/average/good/excellent*” as opposite to a mere *failure/success*
3. estimate base rates by learning class stereotypes

Future work

1. extensive experiments with this formalism
2. extend the formalism to multinomial (complex. mutually exclusive) opinions like ”The level of success is *poor/average/good/excellent*” as opposite to a mere *failure/success*
3. estimate base rates by learning class stereotypes

Thanks for your attention!

Questions?

Thanks for your attention!
Questions?