

Comte

The COM Test Environment

1 Introduction

Comte is a simulation tool to analyze various aspects of the COM theory. The current version of Comte (0.3, but useable) simulates the cycle of message threads in time.

2 Definitionen

$vis_t(n)$: Visibility of message n at time t .

$DS(n)$: Set of *Direct Successors* of n , this is the set of messages directly referenced by n .

$DP(n)$: Set of *Direct Precessors* of n , this is the set of messages, which directly reference n .

$Suc(n)$ (Receptum): Set of *Successors* of n , this is the set of messages referencing n directly or indirectly (transitive closure of $DS(n)$).

$Pre(n)$ (Inceptum): Set of *Precessors* of n , this is the set of messages, which directly or indirectly reference n (transitive closure of $DP(n)$).

3 Visibility Functions

3.1 degree

A degree-based visibility function. The visibility of a message n depends of the number of references from and to n :

$$vis_t(n) = \left(b + \frac{\delta_s |DS(n)| + \delta_p |DP(n)|}{t^\alpha} \right)^\beta$$

b is the basic visibility given to each message, δ_s and δ_p control the influence of $|DS(n)|$ respectively $|DP(n)|$.

α gives the speed of aging and β increases the contrast: messages with low visibility are suppressed more with larger β .

3.2 constant-average

The flow of visibility within the thread. The basic idea here is, that an initial visibility is given to every new message and these visibilities are distributed between the messages according to the graph structure (therefor the average visibility of all messages is kept constant).

$$\begin{aligned}
 \text{vis}_0(n) &= \text{vis}_0 \\
 \text{vis}_t(n) &= \text{vis}_{t-1}(n) && \text{for } |\text{DP}(n)| = |\text{DS}(n)| = 0 \\
 \text{vis}_t(n) &= (1 - \beta\alpha) \text{vis}_{t-1}(n) + \beta\alpha \sum_{m \in \text{DP}(n)} \frac{\text{vis}(m)}{|\text{DS}(m)|} && \text{for } |\text{DS}(n)| = 0 \\
 \text{vis}_t(n) &= (1 - \beta(1 - \alpha)) \text{vis}_{t-1}(n) + \beta(1 - \alpha) \sum_{m \in \text{DS}(n)} \frac{\text{vis}(m)}{|\text{DP}(m)|} && \text{for } |\text{DP}(n)| = 0 \\
 \text{vis}_t(n) &= (1 - \beta) \text{vis}_{t-1}(n) + && \text{otherwise.} \\
 &\beta \left(\alpha \sum_{m \in \text{DP}(n)} \frac{\text{vis}(m)}{|\text{DS}(m)|} + (1 - \alpha) \sum_{m \in \text{DS}(n)} \frac{\text{vis}(m)}{|\text{DP}(m)|} \right)
 \end{aligned}$$

Each message keeps $1 - \beta$ of its visibility and distributes *beta* to its neighbours, where α is distributed to $\text{DP}(n)$ and $1 - \alpha$ to $\text{DS}(n)$.

3.3 pagerank-old

The PageRank visibility function known from Google applied to Comte.

$$\text{vis}_t(n) = b + \frac{\alpha}{N} + (1 - \alpha) \sum_{m \in \text{DP}(n)} \frac{\text{vis}_{t-1}(m)}{|\text{DS}(m)|}$$

N is the total number of messages, b the basic visibility¹ and $1 - \alpha$ the amount given by $\text{DP}(n)$.

3.4 add-aging

Simple aging visibility function (additive).

$$\begin{aligned}
 \text{vis}_0(m) &= \text{vis}_0 \\
 \text{vis}_t(n) &= (1 - \beta)e^{-\alpha t} \text{vis}_0 + \beta \sum_{m \in \text{DS}(n)} \text{vis}_t(m)
 \end{aligned}$$

¹we had to add this because otherwise we would start with zero visibility.

3.5 add-aging-old

Simple aging visibility function (additive).

$$\begin{aligned}\text{vis}_0(m) &= \text{vis}_0 \\ \text{vis}_t(n) &= (1 - \beta)e^{-\alpha t} \text{vis}_0 + \beta \sum_{m \in \text{DS}(n)} \text{vis}_{t-1}(m)\end{aligned}$$

3.6 mult-aging

Simple aging visibility function (multiplicative).

$$\begin{aligned}\text{vis}_0(m) &= \text{vis}_0 \\ \text{vis}_t(n) &= e^{-\alpha t} \text{vis}_0 \cdot \sum_{m \in \text{DS}(n)} \text{vis}_t(m)\end{aligned}$$

3.7 mult-aging-old

Simple aging visibility function (multiplicative).

$$\begin{aligned}\text{vis}_0(m) &= \text{vis}_0 \\ \text{vis}_t(n) &= e^{-\alpha t} \text{vis}_0 \cdot \sum_{m \in \text{DS}(n)} \text{vis}_{t-1}(m)\end{aligned}$$

4 Distribution Functions

4.1 SimpleScaleDistribution

Creates minMsg to maxMsg new messages in each cycle, each with minRef to maxRef references.

The probability of a message to be referenced is proportional to its visibility.