



An Automatic Layout Approach for iStar Models

Xiaolin Du, Tong Li, and Dan Wang
Beijing University of Technology, Beijing, China

The Tenth International i* Workshop (iStar2017)
Essen, Germany
June 13th, 2017



Motivation

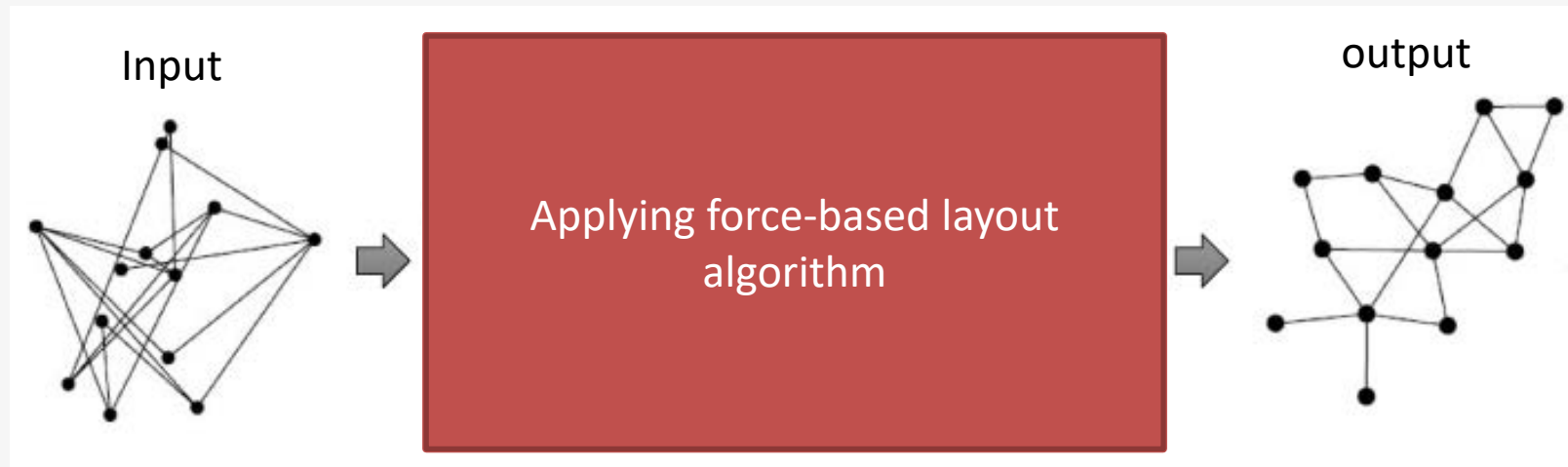


Motivation

- The comprehensive syntax of iStar modeling language allows requirements analysts to clearly capture stakeholder's needs, as well as dependencies among stakeholders.
- Constructing and adjusting iStar models are **laborious** tasks, especially when dealing with **large-scale** models.
- We propose a tentative approach to automatically lay out iStar models using **a force-based layout algorithm**.

Force-based Layout Algorithms

- Force-based layout algorithms simulates a physical system with edges acting as springs and nodes as repelling objects





A Forced-Based Layout Algorithm for iStar Models »

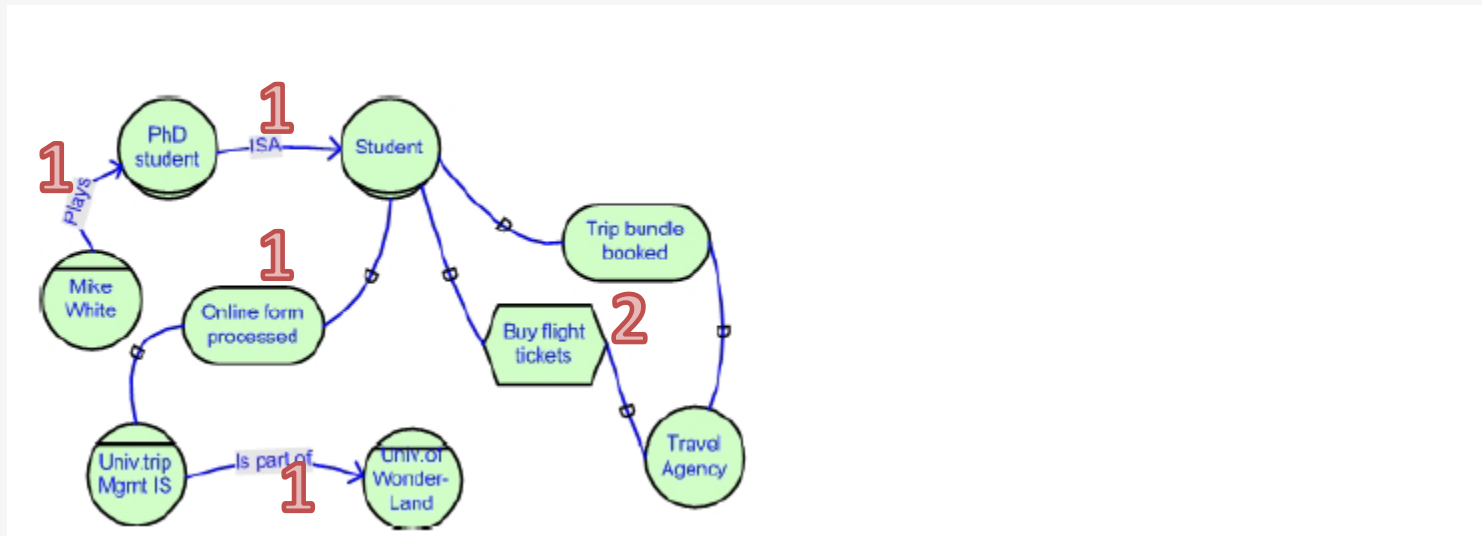
Layout of SD View

- **Objective**

- Calculate a reasonable initial position for each actor.

Model Pre-Processing

- Transform an iStar SD model into an undirected relational graph.
 - Edge Weight Matrix $|EW_{ij}|$



Applying FR Algorithm

- $G(V_G, E_G)$ denotes a relational graph
 - V_G represents the node set of G
 - E_G represents the edge set of G
- Employ FR (Fruchterman and Reingold) layout algorithm to calculate the positions.

Applying FR Algorithm

- Each iteration of the FR layout algorithm needs to (re-)adjust three parameters:
 - 1) Calculate the effect of **attractive** forces on each vertex;
 - 2) Calculate the effect of **repulsive** forces;
 - 3) Limit the total displacement.

Calculation of Forces

- Calculate **radius of the empty area around a node (k)**

$$k = C * \sqrt{\frac{\text{area}}{\text{number of nodes}}}$$

- Given d the distance between the two nodes, attractive forces f_a and repulsive forces f_r are defined as below:

$$f_a(d) = \frac{d^2}{k} \quad f_r(d) = -\frac{k^2}{d}$$

Customized Calculation of Forces

- Takes into account the weight of each edge

$$f_a(d) = EW * \frac{d^2}{k}$$

- A threshold τ is defined to control the minimal distance between connected nodes.

Layout of SD View

Algorithm 1 A customized force-based layout algorithm

Input: a textual specification of a iStar model

Output: a graphical representation of the iStar model

- 1: Transform the iStar model to a relational graph G ;
- 2: Calculate an edge weight matrix EW ;
- 3: Apply the improved FR algorithm;
- 4: { Calculate k
- 5: **For** $i = 1$ to iterations **Do**
- 6: Calculate repulsive forces;
- 7: Calculate attractive forces;
- 8: Adjust the total displacement;
- 9: if the distance between two nodes is less than τ
- 10: adjust moving direction;
- 11: }
- 12: Construct the graphical iStar model

Layout of SR View



- Two sub-tasks (future work):
 - Laying out all the intentional elements inside actor boundaries (a tree-like structure)
 - Positioning each actor boundary as a whole in the entire picture

Evaluation Plans

- Can our approach create understandable layout of iStar models?
- Is our approach scalable to large-scale models?
- Can our approach lead to faster modeling practices?



Conclusions Future Work



Conclusions

- A tentative approach to automatically lay out iStar models using a customized force-based layout algorithm
 - Accelerating iStar modeling practices
 - Helping modelers to focus on the content of models
 - (Eventually) Promoting the adoption of iStar

Future Work

- Extend our proposal to deal with automatic layout of the SR view
- Implement our layout approach in a particular iStar modeling tool
- Empirically evaluate our approach



Thanks!



Q&A

