PAPER *Special Section on Knowledge, Information and Creativity Support System*

Method for Visualizing Complicated Structures Based on Unified Simplification Strategy

Hiroki OMOTE†, ***Nonmember and*** **Kozo SUGIYAMA**†a), ***Member***

SUMMARY In this paper, we present a novel force-directed method for automatically drawing *intersecting compound mixed graphs* (ICMGs) that can express complicated relations among elements such as adjacency, inclusion, and intersection. For this purpose, we take a strategy called *unified simplification* that can transform layout problem for an ICMG into that for an undirected graph. This method is useful for various information visualizations. We describe definitions, aesthetics, force model, algorithm, evaluation, and applications.

key words: *information visualization, graph drawing, intersecting compound mixed graph, force-directed method*

1. Introduction

Many methods for automatic graph drawing so far have been developed to nicely draw rather simple classes of graphs such as trees, planar graphs, directed graphs, and undirected graphs [1]. Recently, it is often needed to treat complicated structures in information visualization, visual editing and so on. One of recent research directions in graph drawing area is to draw more complicated graphs: *mixed graphs* [2] where two types of adjacencies (i.e. directed and undirected edges) are admitted, and *compound digraphs* [3] and *clustered graphs* [4] where inclusion relations (or nesting) among vertices are allowed as well as adjacency. Further, intersection relations among vertices are introduced in an *intersecting clustered graph* [5]. Classification of graphs and visual representations of them are shown in Fig. 1. An *intersecting compound mixed graph* (ICMG) can express more complicated information and knowledge structures. Examples of this include KJ diagram [6] in creativity science (see Fig. 2), Higraph [7] in software engineering, Conceptual graph [8] in knowledge engineering, and Web ontology [9] in the Internet engineering.

As drawn objects become more complicated, drawing techniques also become more complex. Theoretical algorithms (e.g. orthogonal drawing [1]) can attain aesthetic criteria exactly, but it is often difficult to understand and implement them. On the contrary, heuristic algorithms (e.g. force-directed drawing [1]) cannot attain aesthetic criteria exactly, but are relatively easier to understand and implement them. This ease of use is especially important for the user who wants to develop and improve drawing tools by oneself.

In this paper, we develop a heuristic method to draw a graph called an ICMG where adjacency, inclusion, and intersection among vertices, and both types of directed and undirected edges are allowed. We adopt force-directed placement techniques based on simulation of a virtual physical system. These techniques also are suitable for interactively editing diagrams with preserving the user's *mental map* in an interactive environment. As one of the early works for drawing an undirected graph, Eades [11] proposed

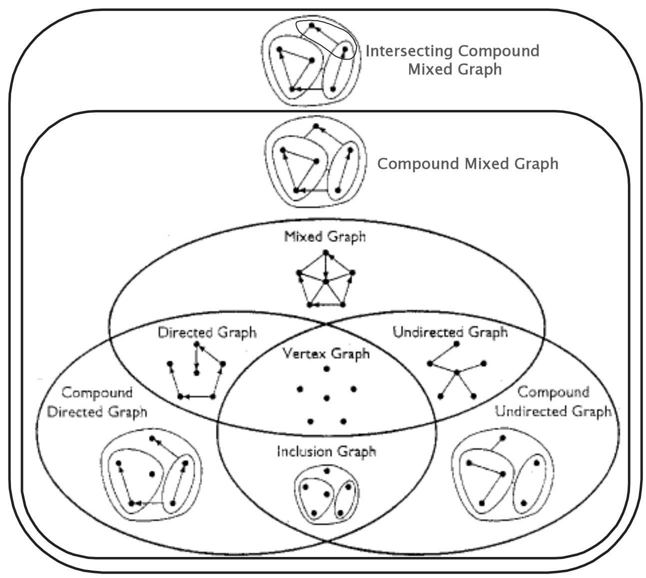


Fig. 1 Classification of graphs.

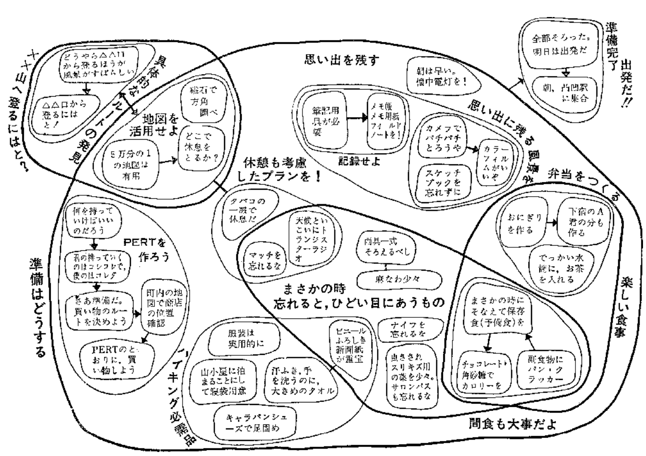


Fig. 2 Example of complicated structure: KJ diagram that is used for idea organizing [10].

Manuscript received February 13, 2007.
Manuscript revised April 23, 2007.
†The authors are with JAIST, Nomi-shi, 923–1292 Japan.
a) E-mail: sugi@jaist.ac.jp
DOI: 10.1093/ietisy/e90–d.10.1649

the spring embedder, though similar idea was previously discovered in [12], [13]. We extend this into a new method by introducing several types of forces to treat more complicated structures. It looks very complicated but actually we can easily implement and modify the method. Our method is characterized by:

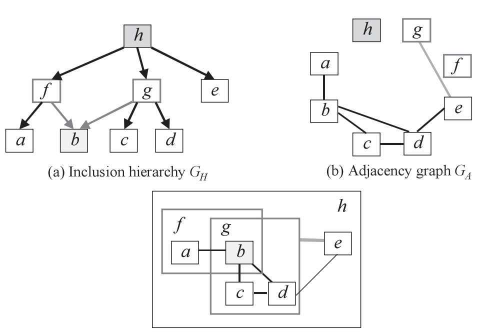
- *Unified simplification* of layout problem for a given complicated graph by transforming it into layout problem for an undirected graph,
- *Parameter tuning* for coping with the increase in the number of parameters controlling forces due to the transformation, and
- *Scheduling* so as to switch the exertion of forces on/off.

In the subsequent sections, we describe definitions, aesthetics, model and algorithm, performance evaluation, and applications.

2. Preliminaries

2.1 Intersecting Compound Mixed Graph

Referring to Sugiyama and Misue [3], when two kinds of binary relations, inclusion and adjacency relations, are defined on a finite set V of vertices, we can introduce two specific graphs to the relations. An inclusion hierarchy (or rooted multi-level directed graph) is a pair $G_H = (V, E)$ where E is a finite set of inclusion edges whose element $(u, v) \in E$ means that u includes v. A vertex including no vertex is called a *leaf*, and a vertex including vertices is called a *cluster*. An adjacency graph is a pair $G_A = (V, F)$ where F is a finite set of adjacency edges whose element $(u, v) \in F$ means that u and v are adjacent. In G_A, element $(u, v) \in F$ usually undirected but can be directed. An ICMG is defined as a triple $G_I = (V, E, F)$ obtained by compounding these two graphs (see Fig. 3). For simplicity, it is provided that every element in F is undirected in this paper.



(a) Inclusion hierarchy G_H (b) Adjacency graph G_A

$G_I = (V, E, F)$,
$V = \{a, b, c, d, e, f, g\}$,
$E = \{(f, a), (f, b), (g, b), (g, c), (g, d)\}$,
$F = \{(a, b), (b, c), (b, d), (c, d), (d, e), (g, e)\}$

(c) Diagram and text expression of G_I obtained from (a) and (b).
(Here root h is omitted in G_I because it is an outer frame in drawing.)

Fig. 3 Example of an ICMG.

2.2 Aesthetics

Aesthetics is a set of drawing conventions and rules. Drawing conventions are conditions that should be exactly satisfied, and drawing rules are conditions that are satisfied as much as possible. We adopt the following two simple drawing conventions:

- *Rectangle* (*c1*): a vertex is drawn as a rectangle or an ellipse (i.e. vertex area);
- *Straight-line* (*c2*): an edge is drawn as a segment of a straight line.

In addition, we adopt five rather complicated rules to draw an ICMG nicely as follows (in priority order):

- *Vertex-inclusion* (*r1*): a vertex belonging to a cluster is drawn within the area of the cluster, and a vertex not belonging to clusters (except the root) is drawn outside the clusters;
- *Vertex-overlap-reduction* (*r2*): undesirable overlapping of vertices (except intersecting vertices) is reduced;
- *Vertex-even-distribution* (*r3*): Vertices are distributed evenly;
- *Vertex-closeness* (*r4*): adjacent vertices are placed closely but not too closely; and
- *Edge-crossing-reduction* (*r5*): crossings of edges are reduced.

The last three rules (*r3*, *r4*, *r5*) are generally accepted and succeeded in the forth-directed placement for general undirected graphs (for example, see [14]). The former two rules (*r1*, *r2*) are specific for this study: if we assign these two rules as drawing conventions instead, solving such a layout problem becomes very complicated like the drawing method for compound directed graphs appeared in [3]. Therefore, we concern the former two rules in the evaluation of our method although it is incidentally successful in satisfying the last three rules.

3. Drawing Method

3.1 Unified Simplification of Layout Problem

We show how to transform a given complicated graph into an undirected graph for the sake of unified simplification of layout problem for an ICMG.

First, we introduce three kinds of dummy vertices for each cluster in a given graph. Each cluster is replaced with a dummy vertex that is placed at the center of the cluster area. Second, each side of a rectangle corresponding to the cluster is replaced with a dummy vertex for a vertical side or horizontal side as shown in Fig. 4.

In order to adjust the width and height of a cluster, we introduce two kinds of dummy edges between dummy vertices for horizontal sides of a cluster, and between dummy vertices for vertical sides of a cluster (see Fig. 5). The lengths of these edges control the size of a cluster.

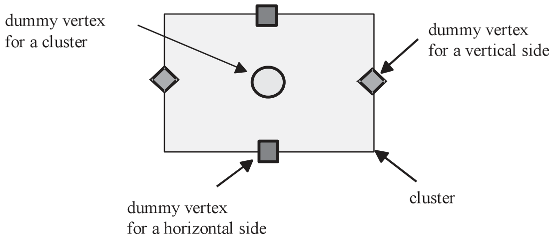
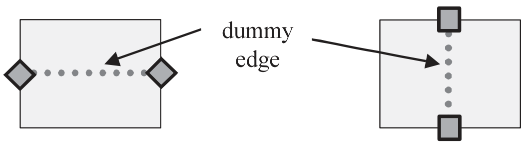


Fig. 4 Three kinds of dummy vertices.



(a) Horizontal dummy edge ($\varepsilon 1$). (b) Vertical dummy edge ($\varepsilon 2$).

Fig. 5 Two kinds of dummy edges for a cluster.

In order to layout intersection between clusters, we prepare two ways: (1) dummy edges are introduced between a vertex shared by clusters and dummy vertices for sides of the clusters, and (2) the intersection is separated as a dummy cluster that is connected to its each mother cluster with a dummy edge (see Fig. 6). Further, a dummy edge is added between intersecting clusters. The transformation of an inclusion relation by adding dummy edges is shown in Fig. 7.

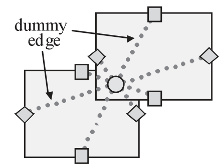
As the results, we have a new undirected graph $G = (V', E')$ corresponding to a given ICMG, where V' called a set of elements and E' a set of links. This introduction of dummy vertices and dummy edges can transfer our layout problem for an ICMG into a unified force-directed drawing problem for an undirected graph. For example, newly obtained undirected graph G corresponding to graph G_I in Fig. 3 is:

$$G' = (V', E')$$
$$V' = \{a,b,c,d,e,f,g,f_d,g_d,f_{v1},f_{v2},f_{h1},f_{h2},g_{v1},g_{v2},g_{h1},g_{h2}\}$$
$$\begin{aligned} E' = \{&(a,b),(b,c),(b,d),(c,d),(d,e),(e,g),(f,g),(f_d,a),\\ &(g_d,c),(g_d,d),(f_{v1},f_{v2}),(f_{h1},f_{h2}),(g_{v1},g_{v2}),\\ &(g_{h1},g_{h2}),(f_{v1},b),(f_{v2},b),(f_{h1},b),(f_{h2},b),(g_{v1},b),\\ &(g_{v2},b),(g_{h1},b),(g_{h2},b),(f_{v1},f_d),(f_{v2},f_d),(f_{h1},f_d),\\ &(f_{h2},f_d),(g_{v1},g_d),(g_{v2},g_d),(g_{h1},g_d),(g_{h2},g_d)\}. \end{aligned}$$

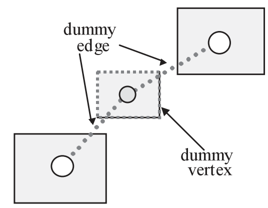
3.2 Forces

To develop a force model, we introduce three types of forces f_s, f_a, and f_r exerted on a pair of elements. If these forces are positive, they are repulsive, and negative attractive. These forces are defined by

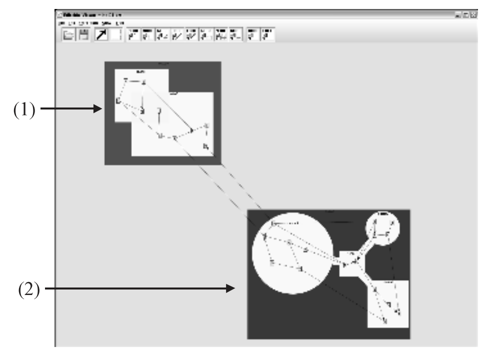
$$f_s = -C_s \log(d/l_s) \tag{1}$$
$$f_a = -C\,d \tag{2}$$
$$f_r = \begin{cases} C\,d & \text{if} \quad d < l_r \\ 0 & \text{if} \quad d \geq l_r \end{cases} \tag{3}$$



(a) Dummy edges for an intersection ($\varepsilon 3$).



(b) Dummy edges and vertex for an intersection ($\varepsilon 4$).



(c) Display images of both ways (1) and (2).

Fig. 6 Two ways of transformations of intersection relations.

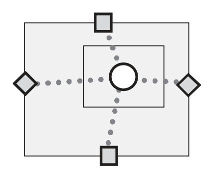


Fig. 7 Transformation of an inclusion relation ($\varepsilon 5$).

where C_s, and C are positive coefficients, and d is the distance between paired elements on which force is exerted. Force f_s is similar to a spring-type force since there exists the concept of ideal distance l_s between paired elements. Force f_s is repulsive when $d < l_r$, and attractive when $d \geq l_r$. Thus, this force can control the distance between paired elements so as to be close to the ideal distance. Force f_a is always attractive (i.e. negative), and varies linearly according to d. Similarly, force f_r is a linear function of d but it is repulsive (i.e. positive) when $d < l_r$ or zero when $d \geq l_r$.

The type of force which is defined for a pair of elements depends on the kinds of elements paired (i.e. leaves, dummy vertices for clusters, and/or dummy vertices for sides of clusters) and the relations between them: for example, force f_s is exerted between adjacent leaves.

Let $p = (e_1, e_2|r)$ be a pair of elements where e_1 and e_2 specify one kind of element and r a kind of relation. We can distinguish four kinds of elements (see Figs. 4–6):

Le: leaf

Dc: dummy vertex for a cluster
Dv: dummy vertex for a vertical side of a cluster
Dh: dummy vertex for a horizontal side of a cluster

Further, we can identify eleven kinds of relations between elements:

- $\rho 1$: between non-adjacent leaves
- $\rho 2$: between adjacent leaves
- $\rho 3$: between a leaf included in a cluster and a leaf not included in the cluster
- $\rho 4$: between two leaves included in a cluster
- $\rho 5$: between a leaf included in a cluster and a leaf included in another cluster where both clusters do not intersect
- $\rho 6$: between a leaf included in a cluster and a leaf included in another cluster where both clusters intersect, but both leaves are not shared by the clusters
- $\rho 7$: between adjacent leaves shared by two clusters
- $\rho 8$: between a leaf and a cluster where both are non-adjacent
- $\rho 9$: between a leaf and a cluster where the leaf is included in the cluster
- $\rho 10$: between non-adjacent clusters
- $\rho 11$: between intersecting clusters

These are shown in the second and third columns of Table 1 formally and schematically, respectively. Thus, we have $e_i \in \{Le, Dc, Dv, Dh\}$, $i = 1, 2$, and $r \in \{\varepsilon 1, \varepsilon 2, \varepsilon 3, \varepsilon 4, \varepsilon 5, \rho 1, \rho 2, \rho 3, \rho 4, \rho 5, \rho 6, \rho 7, \rho 8, \rho 9, \rho 10, \rho 11\}$ for $p = (e_1, e_2 | r)$.

We define a force exerted on every pair of elements according to the specification of pairs presented in Table 1, where schematic representations for relation r are shown in the third column.

3.3 Algorithm

Our algorithm is based on the standard force-directed method [11]. First, the given ICMG G_I is transformed into an undirected graph $G = (V', E')$. Second, a force is defined for each pair of elements according to Table 1. It should be noted that each force has its own parameters used in any one of the force equations (1)–(3), which is one of the characteristics of our method. Though the number of parameters increases, controlling the positions of the elements becomes more flexible and easier.

Next, the direction and magnitude of the total force exerted at each element is calculated, and the position of the element is moved according to its direction and magnitude. Then, given graph G_I is revived, and drawn on the screen. The computational complexity of the algorithm is similar to the original algorithm [11]: $O(|V|^2)$, or higher because the number of iterations tends to become large as $|V|$ becomes large. But this relationship is not evaluated.

algorithm DRAW-ICG (G_I: an ICMG, n: the number of repetitions);

1. Transform G_I into G and define forces on each pair of elements;
2. Place elements of G in random;
3. Repeat n times
   - 3-1) Calculate the force exerted on each element by combining forces based upon force specification given in Table 1 and phase specification shown in Fig. 8;
   - 3-2) Move each element by $\delta\times$(force on the element);
4. Revive G_I from G;
5. Draw the graph on a screen.

Table 1 Classification of relations, forces, and rules.

Force ID	Pair specification $p = (e_1, e_2 \| r)$	Schematic presentation r	Force type
S1	$(Le, Le \| \rho 2$ or $\rho 4)$	adjacent relation or	Spring force f_s
S2	$(Le, Le \| \rho 5)$		
S3	$(Le, Le \| \rho 3$ or $\rho 6)$	or	
S4	$(Le, Le \| \rho 7)$		
S5	$(Le, Dc \| \rho 9)$		
S6	$(Dc, Dc \| \rho 11)$		
S7	$(Dv, Dv \| \varepsilon 1)$		
S8	$(Dh, Dh \| \varepsilon 2)$		
A1	$(Le, Dv$ or $Dh \| \varepsilon 3$ or $\varepsilon 4)$	or	Attractive force f_a
A2	$(Dc, Dv$ or $Dh \| \varepsilon 5)$		
R1	$(Le, Le \| \rho 1)$ or $(Le, Dc \| \rho 8)$	no relation or	Repulsive force f_r
R2	$(Dc, Dc \| \rho 10)$		
Others			

3.4 Parameter Tuning

Parameter tuning is an important mechanism for calculation in our force-directed graph drawing method. When the number of elements is expressed as m, there exist $m^2 (= N)$ pairs of elements: $P = \{p_1, p_2, \ldots, p_N\}$. Finite set P is partitioned by the specification shown in Table 1, where each partition is identified with the Force ID. In our method, a force is defined on every pair of elements. Therefore, we need to treat many parameters for C_s, C, l_s, and l_r.

Parameters C_s, C, l_s, and l_r are empirically determined, which is shown in Table 2. The values of C_s and C for forces *S5*, *S6*, *A1*, *A2*, *R1*, and *R2* are set as 0.3 initially, and increased gradually up to 1.0 in the algorithm while drawing rules corresponding to these forces were not well satisfied. For example, for a relation between a leaf and a cluster that includes the leaf (see Fig. 8), if the leaf is not within the cluster area, parameter C_s of the force exerted on the paired elements is increased (this is the case for *S5* in Table 1). For a relation of nesting of clusters, if the relation is not satisfied, parameter C is increased (this is the case for *A2* in Table 1).

The width and height of every leaf is given initially, which define the area for labelling the leaf. The width and height of every cluster is calculated in the algorithm as the lengths of dummy edges shown in Figs. 4, respectively.

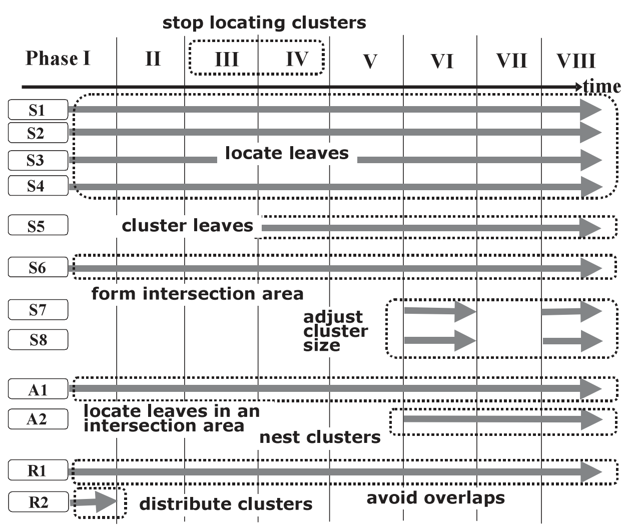


Fig. 8 Scheduling of forces and their purposes.

Table 2 Parameter tuning.

ID	C_s	l_s
S1	0.3	*LENGTH*×2
S2	0.3	*LENGTH*×3
S3	0.3	*LENGTH*×3/4
S4	0.3	*LENGTH*×2
S5	0.3~1.0	*LENGTH*×1/3
S6	0.3~1.0	*LENGTH*×3/4
S7	0.3	*WIDTH*
S8	0.3	*HEIGHT*
ID	C	
A1	0.3~1.0	
A2	0.3~1.0	
ID	C	l_r
R1	0.3~1.0	*LENGTH*/2 (within intersection) or *LENGTH*×2 (others)
R2	0.3~1.0	*LENGTH*×2

In Table 2, *LENGTH* is defined as the maximum value among widths and heights of paired elements, and is used for adjusting parameter l_s. Similarly, *WIDTH* and *HEIGHT* of a cluster control the width and height of the cluster as the ideal distances for spring forces, which are determined so that the cluster area can geometrically cover all the areas for leaves and clusters included by the cluster.

3.5 Scheduling

Scheduling for switching on/off of forces is another important mechanism to control the process to get successful drawings by synchronizing many various forces and eliminating unwilling mutual influences among forces. This can make calculation more efficient and effective than exerting every force at all times. For example, while it is important to attain the distributed and/or intersecting layout of clusters in the initial stage, the fine-tuning of layout such as adjusting the size of clusters becomes more important in the final stage. Based upon this idea, we schedule the on/off switching of forces: when and which force should be exerted or not exerted during the course of computation. Figure 8 shows tentative scheduling in our method, which is empirically developed through trial and error. The time duration for each phase is given as the number of iterations. The scheduling affects the performance of the method significantly. Therefore, more detailed exploration on scheduling is strongly desired as future study.

4. Evaluation

We have made three experiments to evaluate the performance of our method in terms of the following three criteria:

- Undesirable placements for inclusion and intersection (error rate %)
- Undesirable overlapping of clusters (error rate %)
- Undesirable overlapping of leaves (error rate %)

In the experiments, we have randomly generated fifty samples of an ICMG for various cases under the conditions such as: (1) the number of vertices is 100, (2) the mean degree for each vertex is 3, (3) the number of clusters is 2–20, and (4) the number of intersections is 1–10.

Through the experiment, parameter tuning and scheduling are kept as shown in Table 2 and Fig. 8 respectively. Results are obtained after 500 steps in runtime, where

Table 3 Average error rates in the three criteria.

	Criterion		
	Placement of leaves [%]	Overlap of clusters [%]	Overlap of vertices [%]
Experiment 1	0.04	0	0.33
Experiment 2	0.12	0	1.30
Experiment 3	3.83	0	1.70

the number of steps for each phase in the scheduling is fixed as 50–100 through the evaluation.

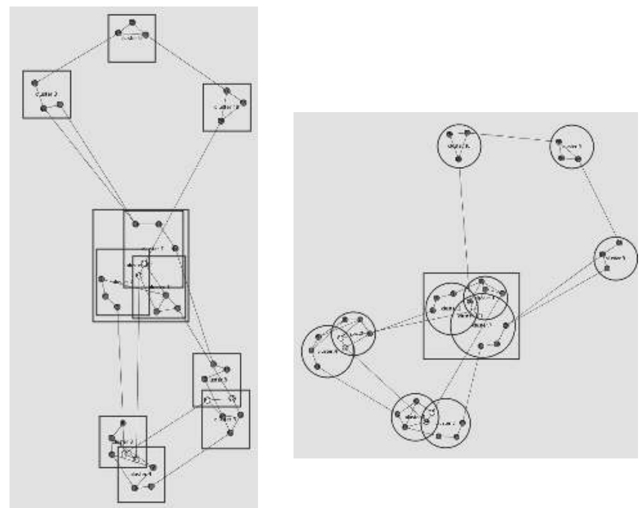
Effects of increase in the numbers of clusters, intersections without nested clusters, and intersections with nested clusters are evaluated in Experiments 1, 2, and 3, respectively. We have the following results (see [5] for more detail):

- The values of the criteria all are satisfactorily low.
- In all experiments, the overlap of clusters is zero.
- We can find increasing tendencies in overlaps of leaves in Experiments 1 and 2, and placement of leaves in Experiment 3.
- Overall average error rates in the three criteria are summarized in Table 3. All values are very small, which means that our method is well satisfied in the performance.

5. Drawing Examples

Figure 9 shows small examples of drawings of graphs. In Fig. 9 (a), leaves are drawn as small circles, and clusters rectangles. The centered cluster includes three subclusters that share two leaves. The inclusion relations between clusters and the relations for sharing both are well drawn in the figure. Figure 9 (b) is another drawing variation of the same graph.

We developed a system for drawing and editing ICMGs. Figure 10 shows a drawing on the screen of our tool: ontology appeared in EURATOM thesaurus [15] is graphically arranged. In the drawing, we can see fifty-five leaves, two intersections, and three-level nesting among clusters and leaves. Figure 11 shows snapshots in interactive



(a) (b)

Fig. 9 Drawing examples.

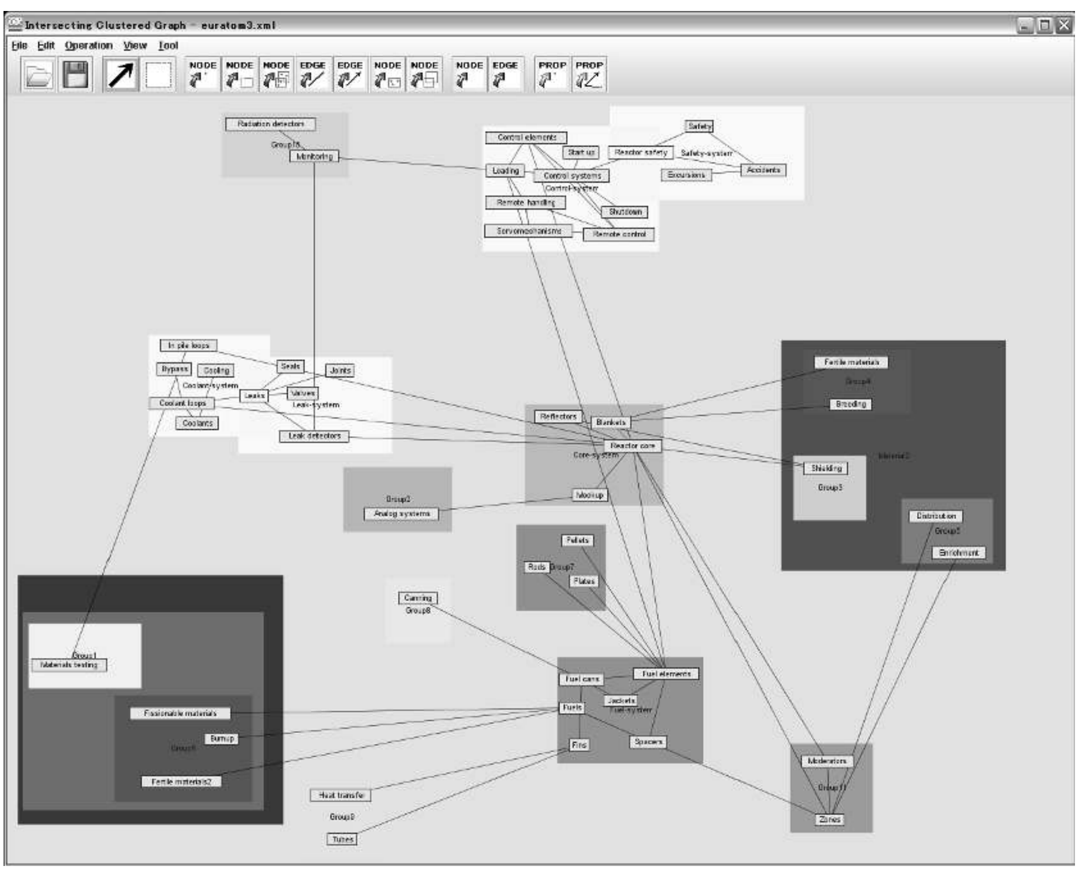
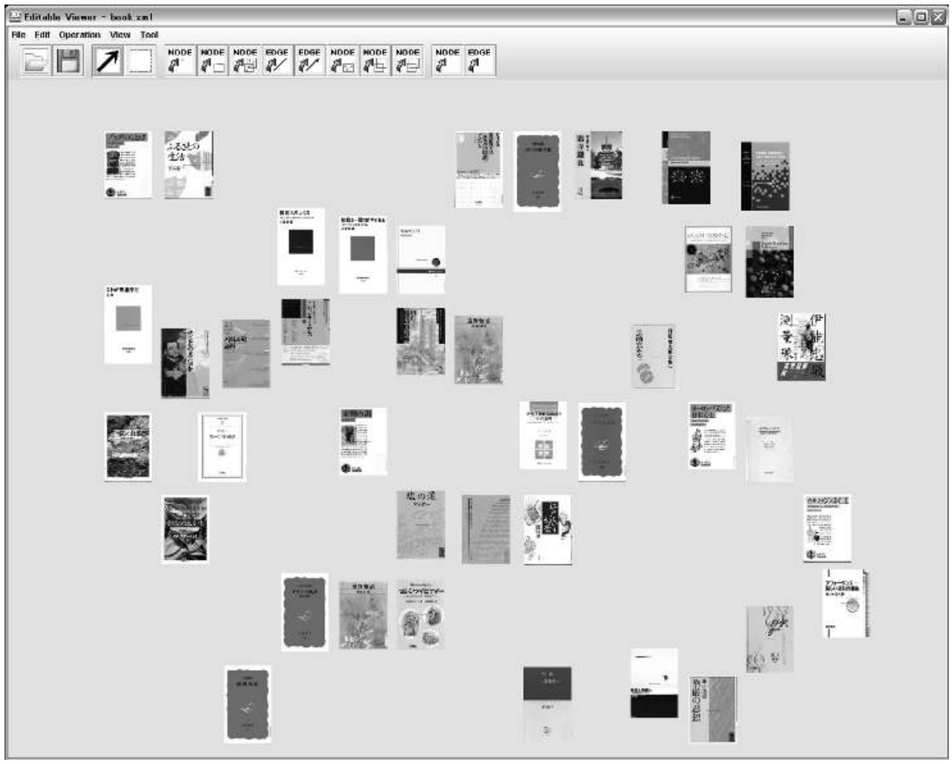
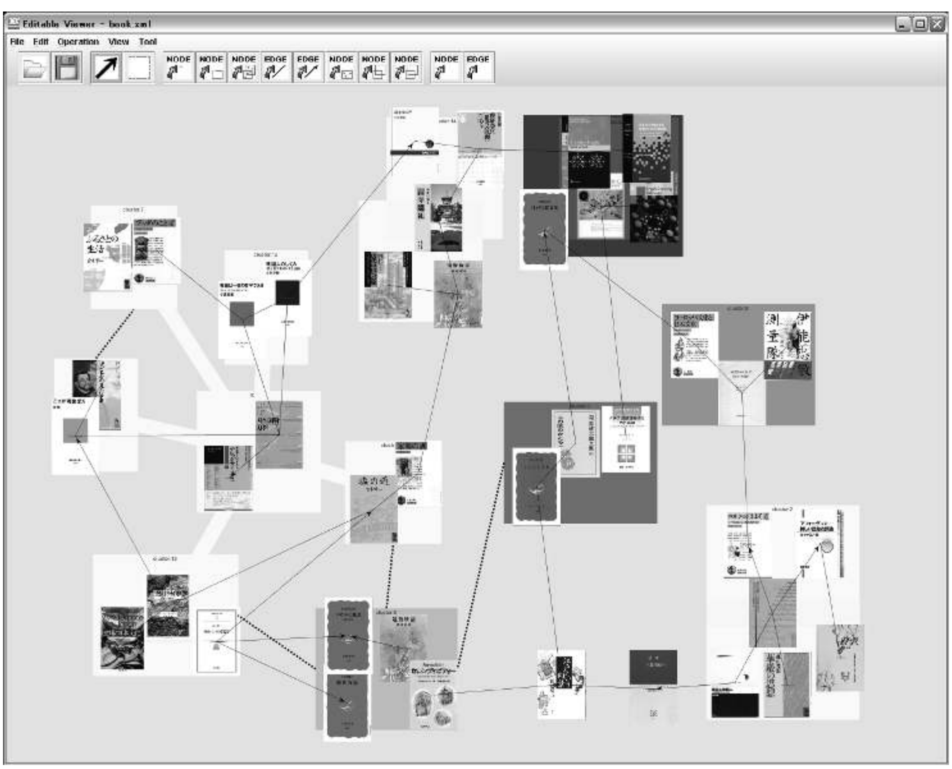


Fig. 10 A real example: Ontology.

visual editing for a real situation: (a) initial drawing and (b) final result, where forty-one books are organized in terms of similarity and user's view. In Fig. 11 (b), two leaves are shared by five clusters. In the drawing, the intersection is



(a) Initial drawing.



(b) Final drawing.

Fig. 11 Snapshots in interactive visual editing for a real work: organizing books.

separated as a new cluster that is connected to the rest part of each mother cluster with an undirected edge, as shown in Fig. 6 (b).

6. Conclusions

In this paper, we have developed a heuristic method for drawing an ICMG where inclusions and intersections are allowed, as well as adjacencies. We adopt force-directed placement techniques to develop this method based on simulation of a virtual physical system. It looks very complicated, but actually we can easily implement and modify the method. Our method is characterized by the techniques called unified simplification, parameter tuning, and scheduling.

Intersections can express elements with several attributes, and clusters can express groupings of similar elements. ICMGs can be used as tools to foster human thought in a variety of fields.

We have described definitions, aesthetics, model and algorithm, performance evaluation, and applications. Specially, we have explained the details of techniques for unified simplification, force definitions, parameter tuning, and scheduling. Results of performance tests show that the values of the three criteria (error rate %) are satisfactorily low. This means that the techniques of our method efficiently perform to create aesthetic drawings for an ICMG.

In the future, we will apply our method to real situations more. Thus our software must be improved in functional usability, for example a provision of an editable graph viewer etc. We will upgrade performance and functional aspects through developing a system. We do not treat controlling the direction of directed edges in this paper. It can be implemented employing the notion of the magnetic spring algorithm [2].

References

[1] K. Sugiyama, Graph Drawing and Applications for Software and Knowledge Engineers, World Scientific, 2002.
[2] K. Sugiyama and K. Misue, "Graph drawing by the magnetic spring model," J. Visual Languages and Computing, vol.6, no.3, pp.217–231, 1995.
[3] K. Sugiyama and K. Misue, "Visualization of structural information: Automatic drawing of compound digraphs," IEEE Trans. Syst., Man Cybern., vol.24, no.4, pp.876–892, 1991.
[4] P. Eades, Q. Feng, and X. Lin, "Straight-line drawing algorithms for hierarchical graphs and clustered graphs," LNCS 1027, pp.113–128, Springer-Verlag, 1996. (Proc. GD'95 Graph Drawing)
[5] H. Omote and K. Sugiyama, "Force-directed drawing method for intersecting clustered graphs," Proc. Asia-Pacific Symposium on Information Visualization 2007 (APVIS2007), pp.85–92, Sydney, Feb. 2007.
[6] J. Kawakita, KJ Method, Chuo-koron-sha, 1986.
[7] D. Harel, "On visual formalism," Commun. ACM, vol.31, no.5, pp.514–530, 1972.
[8] J.F. Sowa, Conceptual Structure*s*, Addison-Wesley, 1984.
[9] C. Fluit, M. Sabou, and F. van Harmelen, "Ontology based information visualization," in Visualizing the semantic web, ed. V. Geroimenko and C. Chen, pp.36–48, Springer, 2002.
[10] J. Kawakita and S. Makishima, Science for Problem Solving, Kodan-sha, 1970.
[11] P. Eades, "A heuristic for graph drawing," CONGRESSUSNUMERANTIUM, no.42, pp.149–160, 1984.
[12] C.J. Fisk, D.L. Caskey, and L.E. West, "ACCEL: Automated circuit card etching layout," Proc. IEEE, pp.1971–1982, 1967.
[13] J.B. Kruskal and J. Seery, "Designing network diagrams," Proc. 1st General Conf. On Social Graphics, U.S. Dept. of the Census, pp.22–50, 1980.
[14] T. Fruchterman and E. Reingold, "Graph drawing by force-directed placement," Software – Practice and Experience, vol.21, no.11, pp.1129–1164, 1991.
[15] B.C. Vickery and A. Vickery, Information Science, in Theory and Practice, Butterworth & Co., 1987.

Hiroki Omote received the M.S. in Knowledge Science from School of Knowledge Science, Japan Advanced Institute of Science and Technology. His main field is information visualization. He is now a doctor student of Japan advanced Institute of Science and Technology.

Kozo Sugiyama received the M.S. and D.S. degrees in Earth Sciences from Nagoya University in 1971 and 1974, respectively. During 1974–1997, he worked for Fujitsu Ltd and Fujitsu Laboratories Ltd to study system engineering, information visualization, and creativity support. He stayed in International Institute of Applied Systems Analysis, Austria in 1982–1983. He is now Professor of School of Knowledge Science, Japan Advanced Institute of Science and Technology.