Memorandum 1835 (April 2007). ISSN 1874-4850. Available from: http://www.math.utwente.nl/publications
Department of Applied Mathematics, University of Twente, Enschede, The Netherlands

Design of a Baggage Handling System

Roxana Grigoraş and Cornelis Hoede

Department of Applied Mathematics
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
{grigorasdr,hoede}@math.utwente.nl

Abstract. In a previous paper we have shown how the design of an object processing system can be reduced to a graph embedding problem. Now we apply the transformations found there to a particular system, namely a Baggage Handling System (*BHS*) of airports, focusing especially on the sorting processors area, as one of the main challenging points. By means of an historical case study, we demonstrate how the method can be successfully applied.

Keywords: Baggage handling system, design process, graph
AMS Classification: 05C99

1 Introduction

For many years it was believed that there is a trade-off between high-quality products and low developing and manufacturing costs, that it is more expensive to develop and create products of higher quality. However, experience and progress in engineering design have proven that lowering costs and increasing quality can be done simultaneously. Next to quality and cost, time comes also as one of the measures of the effectiveness of the design process.
There are several ways of defining the design process. They differ with respect to the purpose and the field one wants to address and apply this process to. Following Ullman's idea about the design process, we can define it as "*the organization and management of people and the information they develop in the evolution of a product*" [Ull]. The progress from the design problem to the final product is performed step by step, by design *decisions*, each of which changes the design *state*. While designing, the initial problem *statement* evolves into an entire framework of knowledge, drawings, models, analysis gathered during this process. The development of the design process between two design states could be viewed as a progressive comparison between a design state and the *goal*, in the case the problem statement is well-defined, so that all the product requirements were correctly specified at the very beginning. *Constraints* could either occur during the design process from the initial requirements, or come as result of design steps/decisions, taken while designing. Every such decision implies two

information units, namely constraints and alternative solutions and an action named *evaluation*, that is confronting constraints with alternatives.
Tools for design are advices, 'rules of thumbs', diagrams, algorithms and so on. Another, more modern, way of presenting them is to integrate these tools in programs for designers, so that elementary procedures of design can be automatised[1]. The rules of thumb do not have to be exactly formulated, they exist mostly in tacit form. A possible reason for this is that these rules could be rather complicated, they can have many exceptions, which makes them difficult to be written down.

In a previous paper [GH], we considered object processing systems and a *Smart Synthesis Tool* (SST) for designing such systems. In this paper we test our theoretical study on the example of a baggage handling system (BHS) of an airport, in which the processed objects are bags.

In Section 2 we describe how a BHS is currently designed in industry, the two first stages of which are the design of a *Process Flow Diagram* (PFD) and a *Material Flow Diagram* (MFD). PFD's will be discussed in Section 3, extending what was considered in [GH]. In Section 4 the application of graph transformations on a PFD, in order to turn it into an MFD, will be investigated for an example. Further aspects of the SST will be discussed in Section 5.

2 Process of designing the Baggage Handling System of an airport

Every project at Vanderlande Industries (VI)[2] can be described by a number of steps, covering the designing, manufacturing and servicing of the system. We will be focusing on the design area; generically, the design of a BHS in VI is performed in three big stages, namely PFD, MFD and Detail Design. In the engineering design process of BHS, a key role is played by the Customer Requirements, as input element in the design chain. Here we have the project requirements, and also the baggage input files, that contain information that helps calculating the baggage flow or the required throughput of the system; in addition, herewith, the system peak flow can be included. From the project requirements, the PFD is constructed.
A PFD gives the connections between the different basic baggage handling processes. To get an idea of the variety of basic processes in a PFD, we considered Fig. 1, where a real-life MFD, for the airport Ankara, is depicted.

Two different elements can be found here, namely *break* and *check-in.* In the 'break' process, the arriving baggage is broken down into 'arrival' and 'transfer'

---

[1] See `http://www2.uiah.fi/projects/metodi/e00.htm`
[2] See `http://www.vanderlande.nl/nl-nl/baggagehandling/Pages/Default.aspx`

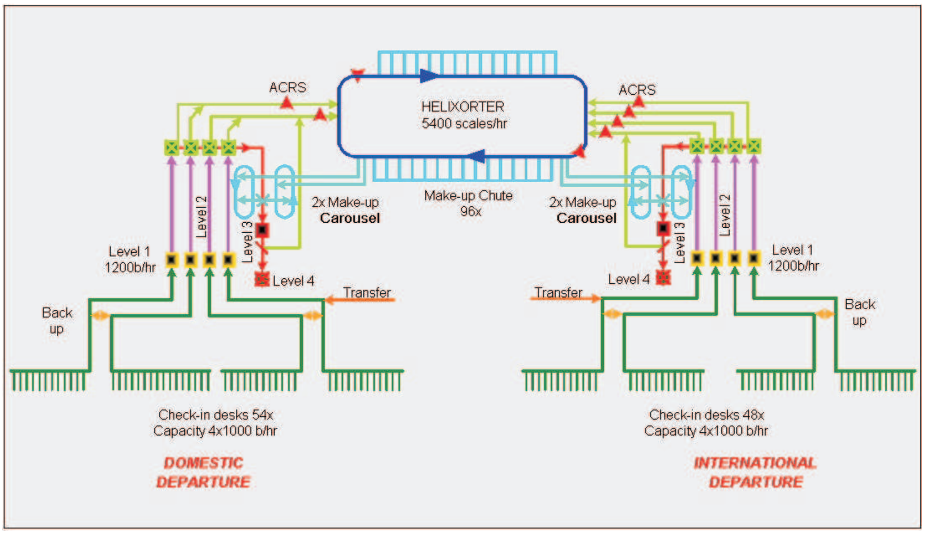

Fig. 1. Example of an MFD

baggage, in 'screened' and 'unscreened' form. 'Arrival' baggage is from passengers who arrived at their final destination and will directly go to the passengers claim area of the airport. 'Transfer' baggage is from passengers from some other airport, who are not yet at their final destination. 'Screened' transfer baggage is already checked, so that there is no need anymore to pass the security area. 'Unscreened' transfer baggage has to go through screening and can be handled as check-in baggage.

Between *check-in* (CI) and *make-up* (MU), there are four different basic processes in the PFD, that is *Early Bag Storage* (EBS), *Identification*, *Sortation* and *Hold Baggage Screening* (HBS). All of these four processes can later be subdivided into decision points and processes handling. At this point, we could define a PFD as being a description of the system in terms of the basic processes a piece of baggage can undergo between CI (or transfer) and exits [Gee].

At CI, agents stick glue-backed bar code labels on baggage, identifying the bag's owner, flight number, final destination and intermediate connections and airlines. The CI agent then puts the bag on a conveyor belt. The number and positioning of CI desks is predefined for every airport, also the number of islands for each desk. Information about the maximum capacity which every CI desk can handle is provided by the customers in the Customer Requirements document. In the EBS the storage of the bags that have to wait for a system exit to be assigned, is made. Not all the airports have an EBS area. It is often the case that the customers of VI want the system to perform without it.

In the Identification process, the correspondent flight, the security status and priority of the bag are determined by an automated scanning station. If its ID-tag cannot be read, the bag is sent to a Manual Coding station, where a human operator scans the bag. The Sortation process directs the bags to the correct system outputs; if the bag cannot be sorted, it is sent back to the Identification area or, if necessary, to the Screening area.

HBS is a very important process where the bags are screened for security reasons. See Fig. 2 for a good illustration of the complexity the basic process HBS can have. In general, five screening levels can be identified. With each level of screening the level of detail with which the bag is examined, increases. In level one screening ($SC1$), the bag is sent through an X-Ray machine and the picture is automatically scanned. If the bag is not cleared, it is sent to the level two screening ($SC2$). At this level, the X-Ray picture previously taken is analysed by a trained human operator. It may be that the bag is directed to another X-Ray machine, where it is again examined. If the bag is not cleared at this stage, it is sent to the level three screening ($SC3$), where a more detailed picture is taken with a CT-scanner. In case the bag is rejected, an operator will examine the picture in the level four screening ($SC4$). If the bag is still suspect, it is sent to the last level of screening ($SC5$), where the bag is removed from the system and manually investigated in the presence of its owner. For simplicity reasons, we will refer to the level of screening as $SC1/2$, $SC3/4$, considering the examination done by the operators in $SC2$ and $SC4$ as being contained in $SC1$, respectively $SC3$.

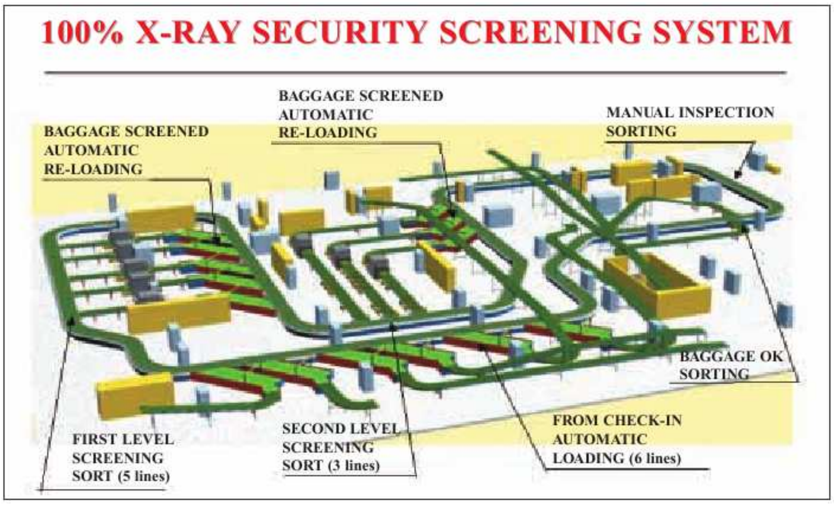

Fig. 2. Security Screening System

Figure 3 gives an example, for the airport Prague, of the way the various basic processes were designed to be interrelated in a PFD.

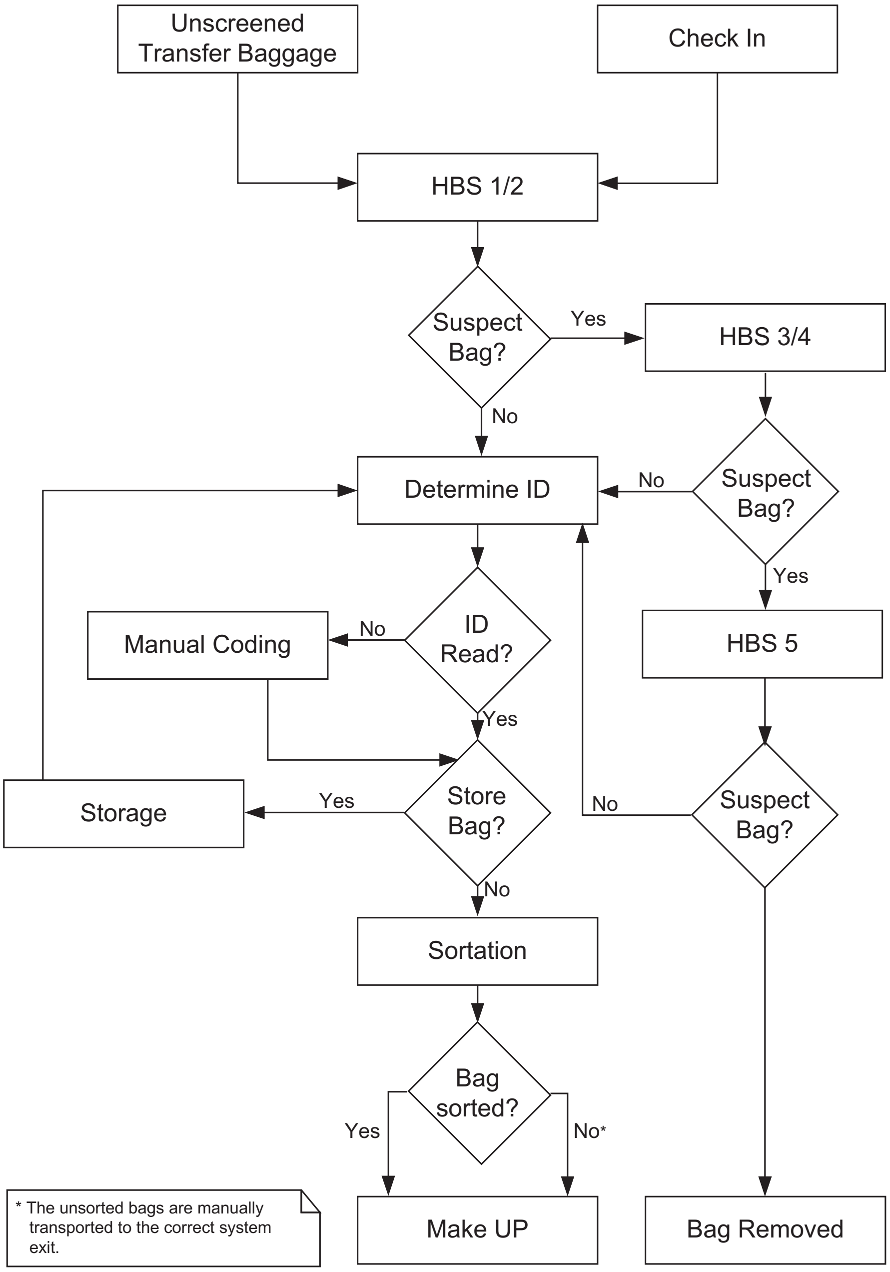

Fig. 3. *PFD* Prague

3 PFD's of a BHS

In [GH] we considered only a PFD in the form of a simple path. In fact, if the only two basic processors are CI and MU, the PFD is as given in Fig. 4.

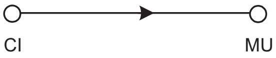

Fig. 4. PFD with only two processors

The basic process of screening HBS introduces the important feature that the flow is splitting due to sortation (SO). Depending on the further chosen processes, e.g. the various levels of screening, such splitting of the flow may occur at different places. As the bags are to end up in the MU area, or to be removed from the system, the PFD, as an undirected graph, see [GH], will show cycles.

One of the first elements of an SST, as conceived in [GH], is the offering of alternative PFD's to the designer. The number of PFD's considered by VI is remarkably low. For our purpose the only important thing to observe is that, as a graph, all PFD's are planar, i.e. can be drawn without crossing of arcs.
We recall that in [GH], in an abstract way, a PFD was to be mapped on a *Geometrical Constraint Graph* (GCG). We considered the problem that the building in which the system was to be implemented was given, i.e. existed. Here we will assume that the airport still has to be built and designed too, together with the BHS. The GCG then just consists of one vertex, on which all vertices of processes of the PFD are mapped. It will be clear that in this case we can skip this consideration and just start with the PFD.
The design of the BHS of Prague airport has started with a PFD, as in Fig. 5, so with a planar directed graph.

We have derived this from the MFD of Prague airport, given in Fig. 6.

We inverted the process of multiplication, as described in [GH]. For example, the 60 CI points, processors, were taken together and "contracted" to one basic process CI, the 3 screening points, processors, were contracted to one basic process $SC1/2$, etc. The symbols in Fig. 5 correspond to parts and areas of the BHS, namely check-in , screening on different levels, transfer, terminal, sorting (different types), manual coding, make-up area, carousel, early bag storage. The small circles represent points where processes are performed, whereas the arrows denote the order in which these take place.

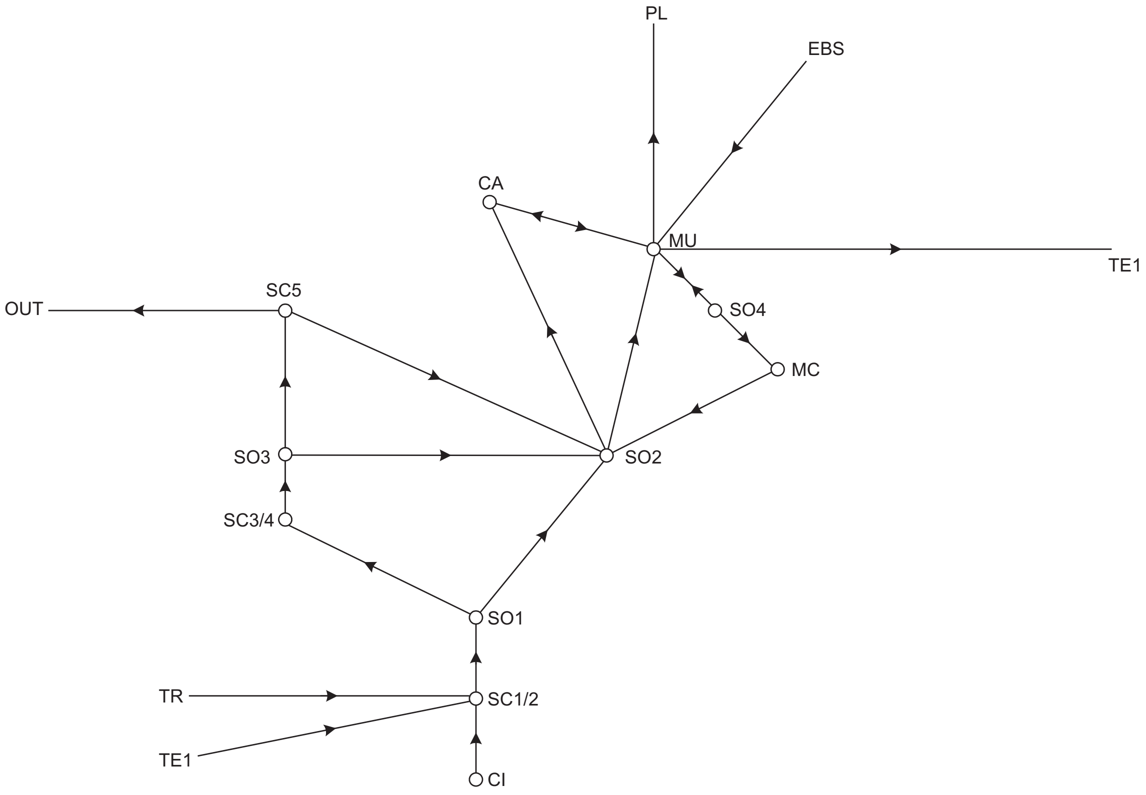

Fig. 5. *PFD* of Prague airport as planar graph

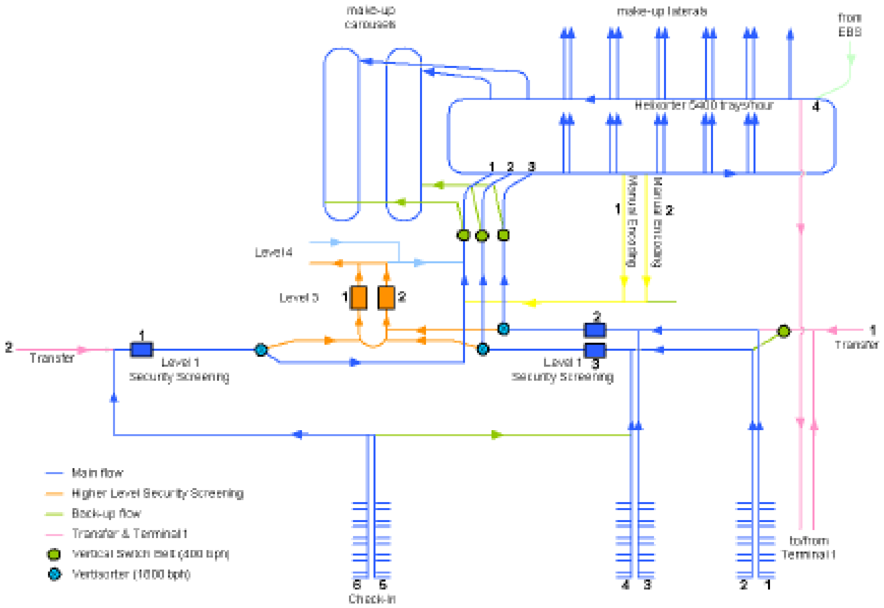

Fig. 6. *MFD* of Prague airport

4 Graph transformations applied on a *PFD*

Our aim in this section is to show how Fig. 5, a PFD, can be transformed into Fig. 6, an MFD, with the help of the SST described in [GH] and that is given in Table 1.

The first step in using the SST has already been described. The PFD is mapped on the one vertex of the GCG. That vertex is then multiplied with 10, because the ten processors have to be located at ten different places. Then the PFD of Fig. 5 is considered again.
The second step is to multiply the processors. The CI vertex is multiplied by 60, the $SC1/2$ processor by 3, the $SC3/4$ processor by 2, the sorting $SO1$ is considered a processor and multiplied by 3, $SO3$ is not multiplied, neither is $SC5$, $SO2$ is multiplied by 3, the carousel CA is multiplied by 2, manual coding MC is multiplied by 2 as is $SO4$.
These factors are determined on the basis of capacity considerations, but the vulnerability aspect is envisaged as well, for which operations research techniques can be used. In Table 1 this is the third procedure.
The third step is the most important one for this paper. For each of the arcs we will now investigate how the transportation could be designed to give the MFD of Fig. 6.

TASK/PROBLEM	**SUPPORT**
Choosing the PFD	The SST offers a survey of possible PFD's from which the designer may choose.
Mapping the PFD on the GCG	The SST calculates all possible mappings of the chosen PFD on the GCG.
Multiplication of the processors	The SST uses a survey of the available pro– cessors, calculates the desired number of pro– cessors and checks whether the chosen multiplication is realisable at the location given by the mapping of the PFD on the GCG.
Placing the processors	The SST offers the designer help in placing the processor at the intended location.
Multiplication of the transporters	The SST can be given a survey of the available transportation systems, then calculates the desired number of transporters and checks whether the chosen multiplication of transporters is achievable.
Generation of layouts	The SST offers a survey of possible layouts of the transportation system, indicating whether the layouts can be realised in a planar way or necessarily involve crossings.
Compactification	The SST offers alternatives for layouts that require less space, due to multi-level design, so that crossings in the designed transportation system can be admitted.

Table 1. Summary of potential support of an SST

We will be focusing in the following on the 'multiplication of the transporters' and the 'generation of layouts' (see Table 1), with emphasis on the fifth procedure. In the next section we will be also considering the 'compactification' aspect.

We recall now briefly the graph transformations T_1, T_2 and T_3, presented in [GH], which will constitute the basis for the following analysis.

The first type of transformation, T_1, describes multiplication of one transportation machine. The transportation is distributed over a certain number of parallel machines. In Fig. 7 this is illustrated, as well as the transition to the total graph form.

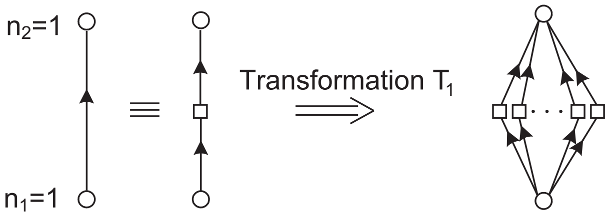

Fig. 7. Transformation T_1. Multiplication of one transportation machine.

The second type of transformation, T_2, describes the multiplication of splitting points and fusion points, indicated by small black circles.

In a general way the two processor groups are partitioned into subgroups, so n_1 processors are partitioned into subgroups of $n_{1,1}, n_{1,2}, \ldots, n_{1,k}$ processors and n_2 processors are partitioned into subgroups of $n_{2,1}, n_{2,2}, \ldots, n_{2,l}$ processors. The basic transportation design is transformed into a design with k splitting points and l fusion points. These are now first to be connected to the processors and then to each other, see Fig. 8.

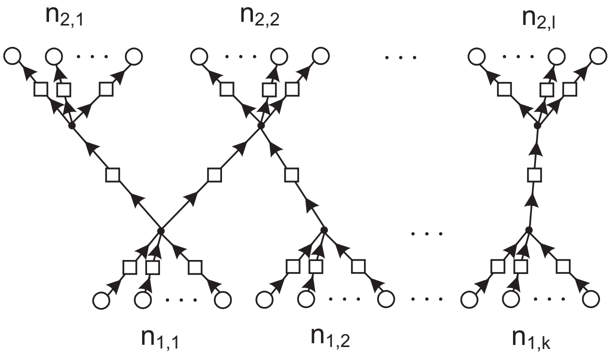

Fig. 8. Transformation T_2. Multiplication of splitting and fusion points.

Figure 9, for example, depicts a situation where having to transport from a single machine to a number n_2 of machines, a possible alternative, from many existing, appears.

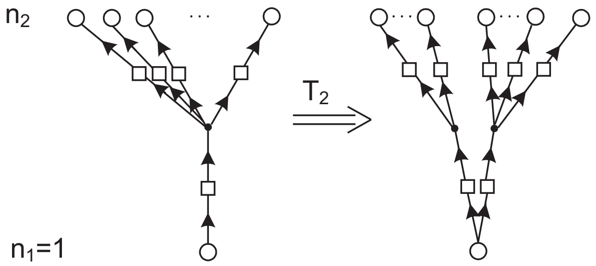

Fig. 9. Multiplication of a splitting point.

Finally, in transformation T_3 two, respectively three transportation machines are replaced by just one transportation machine, see Fig. 10.

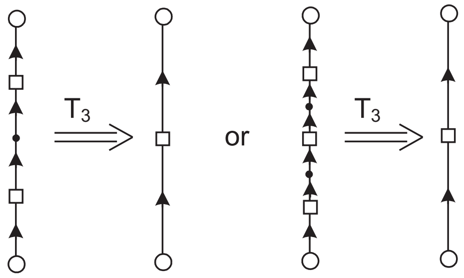

Fig. 10. Transformation T_3. Reduction of successive transportation machines.

Now coming back to our case, consider the path from CI to $SC1/2$ in Fig. 5 (which looks basically like Fig. 4, and that holds for every path between two points, in Fig. 5). The aim is to generate the MFD configuration of Fig. 6, using the transformations T_1, T_2 and/or T_3 developed in [GH]. As it can be seen in Fig. 6, the three big CI points contain in fact a number of sixty check-in desks, twenty for each CI point, arranged parallel in groups of ten's. In fact, those sixty check-in desks have to connect with those three level 1/2 screening machines. As in [GH], we will use as drawing conventions small circles for "splitting points", respectively "points of fusion", and squares for representing transportation machines. Hence, we can design this first path from CI to $SC1/2$ as follows:

a. We represent the arc from CI to $SC1/2$ by sixty arcs from sixty CI points to a fusion point that is connected by one arc to a splitting point, that is connected by three arcs to the three $SC1/2$ points.
b. We apply transformation T_1, which yields multiple (three) transportation machines from the fusion point to the splitting point.
c. We now perform a multiplication with a factor 3 of the fusion point, each of which adjacent to twenty check-in desks, and also a multiplication with the same factor of the splitting point. We apply transformation T_2 by choosing the proper connections. The overall result after applying transformations T_1 and T_2 is the structure of Fig. 11. So, after applying two graph transformations, we have obtained the translation of the simple path between CI and $SC1/2$ (see Fig. 5) as depicted in Fig. 11. In this way, we have derived the first piece of the MFD (Fig. 6).

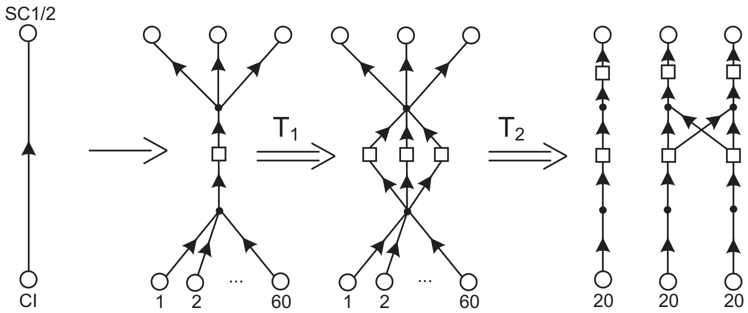

Fig. 11. Partial design of layout

We will proceed now in a similar manner for the arc linking $SC1/2$ and $SO1$ in the planar graph of Fig. 5. On this path, the situation is rather simple, involving just a multiplication of processors with a factor 3. Between level 1/2 screening $SC1/2$ and sorting $SO1$, no ramifications occur, there are three separate branches each of which follows its own way, without any interference, as in Fig. 11, for instance. Thus transformations T_1, T_2 and T_3, applied in this order, will give the picture of Fig. 12.

As a third example of designing a partial layout we discuss the transportation from $SO2$ to CA and MU, as this will bring in the interesting aspect of non-planarity.

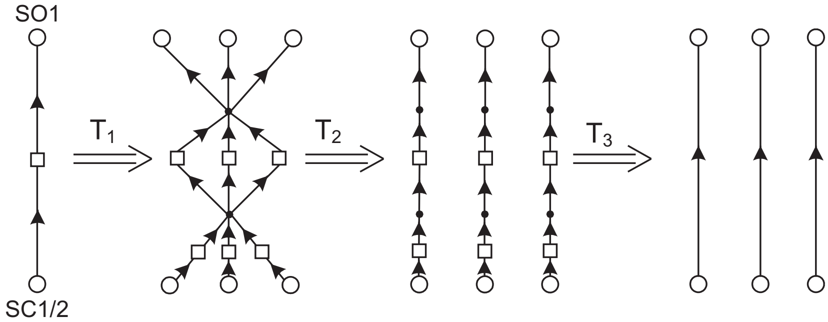

Fig. 12. Partial design of layout

We recall *Kuratowski's Theorem on planar graphs*: *A graph G is planar if and only if G does not contain a subgraph homeomorphic to* $K_{3,3}$ *or* K_5, and the fact that *a graph H is called homeomorphic to a graph G if both graphs can be obtained from the same graph by replacing edges by paths, i.e. subdividing the edges.*

CA consists of two 'ending' points, whereas MU consists of one such point. As $SO2$ consists of three processors that have to be connected to the three processors: CA (two) and MU, in case all $SO2$ are to be connected to all CA and MU, the $K_{3,3}$ structure is present and we necessarily have crossings. In the actual design of the MFD, see Fig. 6, only six transportation processors are present. Therefore first we will be applying the graph transformation T_1, which will lead to a number of six transportation processors. In a next phase, one of these six branches will become a separate arc to one of the CA, two other branches will merge into one arc to the other CA, and the other three will go to MU. The design of the two branches to the CA's, respectively MU will be a bit different: here vulnerability considerations play a role. In Fig. 13 we have applied the transformation T_2 in the following way. Both, the fusion point and the splitting point were multiplied by 2. Now the fusion point towards MU is multiplied by 3 as is the splitting point to MU. This is also an operation T_2. The following operation T_2 involves a multiplication of the fusion point, respectively the splitting towards the CA's by a factor 2, and choosing the proper connections, one $SO2$ to one fusion point and the other two, to a second fusion point, see Fig. 13. The next step illustrates that the designer may decide to reduce the number of transporters. This can be done by the inverse operation T_1^{-1}. Finally we apply transformation T_3 to all of the paths, one of which is from a fusion point to a CA.

From the previous three arcs of the PFD as planar graph of Fig. 5, we have demonstrated how, in a rather unsophisticated manner, the correspondent pieces in the MFD of Fig. 6 could be derived. Since the three cases were representative for the whole picture, that is Fig. 5, we can say that the transformations T_1, T_2 and T_3 suffice and are quite satisfactory, in the sense that the generation of an

MFD does not pose any special problems. The point is that any arc of Fig. 5 could be taken and those three graph transformations could be applied. The order in which these have to be applied may differ for every arc. When doing so, the entire MFD can be generated. The only challenge which remains now is a clever way to implement this method in the SST tool, but this is not our present concern.

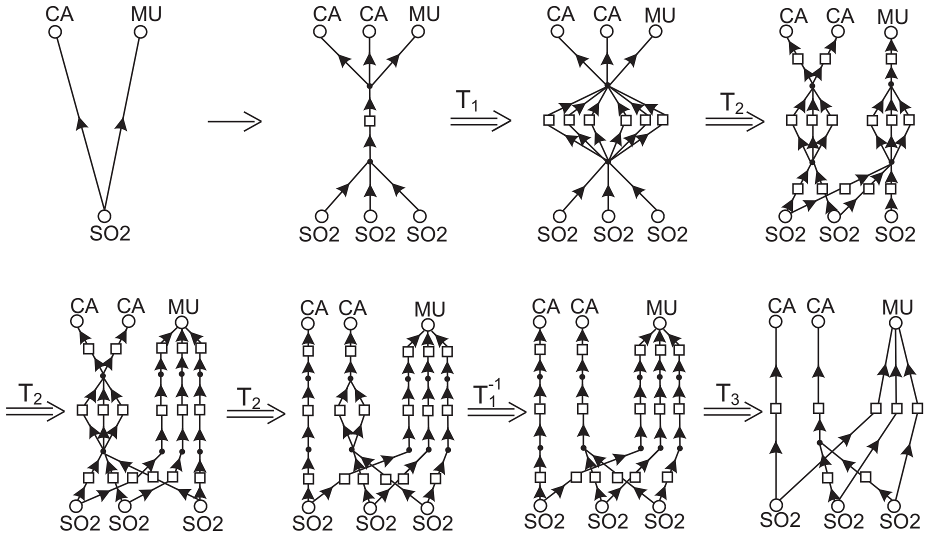

Fig. 13. Partial design of layout

5 On compactification

We have seen that the first step in the design led to a PFD that was planar, see Fig. 5. Due to multiplication of processors and transportation processors, conveyor belts, the designer faces the problem of choosing the connections when applying operation T_2 in which fusion points and splitting points are multiplied as well. As is clear in the second example of Section 4, a simple planar configuration can be chosen for connecting the three $SC1/2$ processors with the three $SO1$ processors. However, the third example showed that the designer can choose a non-planar configuration for connecting the three $SO2$ processors to the two CA and one MU processors, in the case of the Prague MFD.

The point we want to make is that when the building is to be designed together with the BHS one can allow any non-planar layout. If transportation has to take place on one level and crossings of flows have to be avoided, a planar design of the layout is obligatory. If, however, the building is to be adjusted to the layout, crossings of flows can be allowed in the design, as they are handled by letting flows take place on different levels of the building.

Now this seems to be a costly affair as, instead of one big hall, several stories are to be built. However, there are reasons to consider designs that exhibit non-planarity. Suppose the area needed for a planar design is considerably larger than for a non-planar design. This not only means that more ground is needed, that might not be available, but also that the length of the transportation processors would have to be considerably longer. This again would possibly cost more, but, probably more important, may affect the capacity of the transportation processors. So, it might be necessary to accept non-planarity. But then another problem comes in, namely that of *compactification.* It might be a demand to choose an MFD that allows a very compact realisation. The actual realisation of the BHS in Prague is quite compact and knows various levels.

We call *compactification* of a design, a transformation of the design, such that the actual realisation takes as few space as possible. This will involve minimizing distances between processors. This problem is related to the well-known problem of *sphere packing*, where identical spheres are required to be packed in such a way that the "density" of the packing is maximized.

The compactification problem certainly plays an important role in making the final design. To make clear why compactification may be profitable, we consider two groups of three processors that are connected according to the graph $K_{3,3}$. In Fig. 14 we have given two drawings, one with only one crossing and one with eight crossings. The distances are chosen as indicated and just as an example.

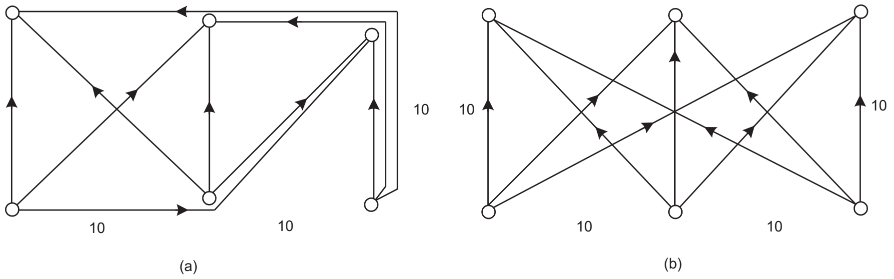

Fig. 14. Two layouts for a $K_{3,3}$ configuration

In Fig. 14(a), there is a transportation processor of length 30 that has length $10\sqrt{5} \approx 22.3$ in Fig. 14(b), one of length 20 that has length $10\sqrt{2} \approx 14.1$ in Fig. 14(b) and one of length $10(1 + \sqrt{2}) \approx 24.1$ that has length $10\sqrt{5} \approx 22.3$

in Fig. 14(b). The other six transportation processors have the same length in (a) and (b). So (b) may be considered to be 'better' in this respect. Moreover, the connections between the elements of the layout in (b) with other processors need not cause crossings with transportation processors from or to those other processors, whereas for the layout of (a) they will, in most cases.

In Fig. 14(b) we see that allowing crossings indeed gives the possibility to compactify the design and the corresponding physical realisation. In Fig. 14(a) fewer crossings concerning the transportation processors between the two groups of consecutive handling processors occur. However, if a group of handling processors is designed to be part of a transportation process with a third group of handling processors, either before or after in the ordering of the PFD, there may be crossings of transportation processors belonging to two different pairs of consecutive handling processors.

The SST may be involved in the following way. First, all transportation processors between two consecutive handling processor groups are designed as in Fig. 14(b). Thus only, though many, crossings of transportation processors occur between consecutive pairs of groups of handling processors in the designed MFD. The graph of the MFD can now be considered as a candidate for a compact design. In case the designer does not want that many crossings in the overall design, the natural problem arises to draw the graph with as few crossings as possible. It is here that the SST can support by calculating the solution. This is not an easy task as the problem is $NP-hard$ in the sense of complexity theory, see [Dic].

6 Discussion

We have shown that the three transformations introduced in [GH] indeed allow the design of the MFD of Prague airport. Implementation of these steps in the SST seems straightforward.
We also pointed out that non-planarity of designed layouts stems from the multiplication process, that takes place in view of capacity and vulnerability considerations.
Finally, we discussed shortly the notion of compactification once non-planarity is accepted, as buildings can have several levels. From a graph-theoretical view the compactification aspect seems to pose the most interesting challenge, as the problem of drawing a non-planar graph with minimum number of crossings is an $NP-hard$ problem.

In the bibliography only a few items are references. The other items are given as general background.

Bibliography

[BR] Braha, D., and Reich, Y., *Topological structures for modeling engineering design processes*, Research in Engineering Design, Vol. 14, (2003), pp. 185–199.

[Car] Carson, J., *Panel on transportation and logistics modeling*, Proceedings of the 1997 Winter Simulation Conference.

[Dic] Dickerson, M., *Visual Drawings: Visualizing Non-planar Diagrams in a Planar Way*, Journal of Graph Algorithms and Applications, Vol. 9, No. 1, (2005), pp. 31–52.

[Die] Diestel, R., (2005). *Graph Theory.* Berlin: Springer.

[Gee] Geerdes, W., *Smart design of baggage handling systems?*, Master Thesis, (2007), University of Twente, The Netherlands.

[GH] Grigoras, D.R., and Hoede, C., (2007) *Design of Object Processing Systems*, Memorandum 1830, Department of Applied Mathematics, University of Twente, Enschede, ISSN 1874-4850.

[JC] Jim, H.K., and Chang, Z.Y., *An airport passenger terminal simulator: A planning and design tool*, Simulation Practice and Theory, Vol. 6, (1998), pp. 387–396.

[LL] Leone, K., and Liu, R., *The key design parameters of checked baggage security screening systems in airports*, Journal of Air Transport Management, Vol. 11, (2005), pp. 69–78.

[PH] Pagani, J., and El Halim, A., and Hassan., Y., and Easa, S. *User-perceived level-of-service evaluation model for airport baggage-handling systems*, Transportation Research Record 1788, (2002), pp. 33–42.

[PR] Pons, D.J., and Raine, J.K., *Design mechanisms and constraints*, Research in Engineering Design, Vol. 16, (2005), pp. 73–85.

[Sal] Salustri, F.A., *Towards an action logic for design processes*, International Conference on Engineering Design Iced 03 Stockholm, August 19-21, (2003).

[Suh] Suh, N.P., *Axiomatic Design Theory for Systems*, Research in Engineering Design, Vol. 10, (1998), pp. 189–209.

[Ull] Ullman, D.G., (1997). *The mechanical design process.* New York: McGraw Hill.

[Van] System Book, Part 2 *Baggage handling*, 2001, Vanderlande Industries, Veghel, the Netherlands.