

Probabilistic Reactive Disassembly Planning

Geiger D., Computer Science Department, Zussmann E., Mechanical Engineering Department,
Technion, Israel Institute of Technology, Haifa, Israel

Submitted by E. Lenz (1)

Received on January 5, 1996

Abstract

A paradigm for reactive disassembly planning is developed which combines prior information about the disassembly task encoded in a predictive plan as well as measurements collected during the process in order to select the current most promising disassembly operation. Planning relies on a probabilistic inference mechanism which employs a Bayesian Network. It integrates perception and action and uses the dependencies among various parts of a product to propagate uncertainty regarding their condition as sensed during the disassembly process. Our methodology allows a computerized assessment of an End-Of-Life value of a product which will enable manufacturers to combine design principles that take into account environmental considerations at minimum cost during the design cycle. The approach is demonstrated by a case study.

Keywords:

Disassembly, Planning, Statistical

1. Introduction

Environmental legislation urges many manufacturers to develop effective technologies to cope with obsolete products. The natural solution is product disassembly which can lead to cost minimization, hazardous materials isolation, and opportunities to re-use or re-utilize materials and components. However, unlike assembly processes, disassembly processes are characterized by a high variety of products and manufacturers, uncertain product condition after usage, and a termination goal that has to be updated during the disassembly process. Therefore, an appropriate methodology must be able to generate plans of disassembly processes that take these features into account.

In this paper we develop a paradigm for *reactive* disassembly planning which combines prior information about the disassembly task encoded in a *predictive* plan as well as measurements collected during the process in order to select the current most promising disassembly operation.

A *predictive* plan takes into account the generic product, its design, materials, mode of usage, expected deterioration, etc., but ignores the conditions of a specific product as revealed during the disassembly process. Given a predictive plan, one can start a disassembly process, however, in general, the plan should be adapted to the current condition of the product. The adaptation is carried out dynamically during the recycling process whenever the information in the predictive plan becomes *incomplete*. This re-planning process is often called *Reactive* planning; it can be viewed as the task of re-establishing a valid plan once the assumptions of a predictive plan need an update.

We present a model-based planner which relies on a probabilistic inference mechanism. Our reactive planner employs a predictive plan which is represented using a *Bayesian Network* [6]. A Bayesian network is a knowledge-based system that represents a collection of random variables combined with Markovian properties of independence. Using this network, one can represent time-variant parameters of the product such as rust and deformation at various locations. Other random variables represent success or failure of specific disassembly operations. This network is called the Static model. In

addition we have a *Recovery Graph* which models the various disassembly options, their costs and their benefits. Once the most promising action is taken, the appropriate random variable in the Bayesian network is set to its observed value (say, success or failure) and one can compute the success probability of each of the next possible disassembly operations using the well developed techniques of Bayesian networks. Then, a new plan is generated using the updated success probabilities such that the End-Of-Life (EOL) value is maximized. To the best of our knowledge, it is the first model for disassembly plans which integrates perception and action and which uses the dependencies among various subsystems of a product to propagate uncertainty regarding their condition as sensed during the disassembly process. Currently, disassembly planners within the Computer Aided Process Planning (CAPP) community assume that the product is known a-priori with certainty [2,7,8].

The rest of this paper is organized as follows: Section 2 introduces the Static Model which describes the probabilistic inference mechanism and introduces the product's Bayesian Network. Section 3 presents the Dynamic Model, which integrates probabilistic inference and operational decision of the disassembly process and yields a reactive planner. A case study is presented in Section 4.

2. The Static Model

The basic premise in building a model for disassembly is the presence of uncertainty in the abilities to disassemble a product up to its basic elements. Deformation, rust, missing parts, unloosenable joints are some of the factors that may influence disassembly and may suggest an appropriate disassembly method, e.g., destructive or non-destructive [1]. Moreover, the ability to disassemble one part of a mechanical system is tightly related to the disassembly of other parts through the topology of the system, e.g., a deformation in one end of an assembly raises the probability that other objects have been deformed as well. A model for disassembly must represent the topology of a product, and use the representation to propagate the effect of disassembly operations (whether successful or not) applied to one part

of the product to other yet-to-be-decomposed object. Clearly, any effect can influence the entire system, however, restricting the effect to neighboring components is a sensible computational compromise. That is to say, we assume a *Markovian* property, in which the disassembly of an object is independent of the decomposability status of other objects, given the status of its immediate neighboring objects.

Let us consider an assembly composed of a chain of objects. Let A_1, \dots, A_n denote these objects and J_2, \dots, J_n denote their joining elements (Fig. 1). We define a set X of random variables $x_i, i = 2 \dots n$ such that x_i is true if joint J_i can be disassembled, and false otherwise. Our Markovian assumption, in this example, states that given values for x_{i-1} and x_{i+1} , x_i is independent of $X \setminus \{x_{i-1}, x_{i+1}\}$. This model summarizes all the factors influencing a particular joint with one bi-valued random variable. Of course, having several values to each x_i to indicate the degree of decomposability, is a reasonable extension.

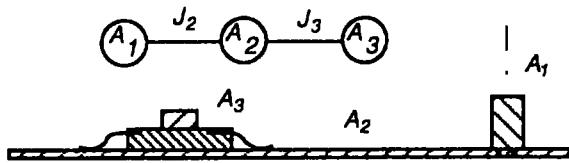


Figure 1: An Electronic Assembly $\{A_1, A_2, A_3\}$ and its Connectivity Graph for $n=3$

Our purpose, however, is to model each individual factor, such as rust and deformation, according to the way it influences disassembly, as well as the way local disassembly attempts influence the whole disassembly process. Moreover, we would like to model sensing operations which may cause further uncertainty as well. Consequently, we extend the Markov chain and use what is known as a *Bayesian Network*. Instead of giving a formal definition we shall first build a Bayesian network for this assembly example. We define a random variable d_i to represent the deformation level of part A_i , and r_i to represent the rust level of part A_i . The domain of d_i and r_i is discrete, as is the domain of all variables we consider. We define j_i as a random variable that connotes the hardness of the connection between A_i and A_{i-1} , that is, the degree of difficulty of separating these objects. Variable s_{ij} stands for "joint i can be disassembled using disassembly method j ". Finally, f_{ik} is the result of a sensing operation k , applied after a disassembly attempt has been performed on joint J_i . A graphical model of the Bayesian Network of the above assembly is given in Figure 2.

Note that edges are drawn from causes to their effects whenever possible. For example, r_i, d_i, r_{i-1} and d_{i-1} are the only factors that influence the hardness of decomposability of joint i . Also, the success of a disassembly operation depends only on j_i , the hardness of the joint, and sensing depends only on the decomposition method used. The joint distribution of the variables relevant to the i 's joint as implied by this model is given by:

$$P(d_i, r_i, j_i, s_{i1}, s_{i2}, f_{i1}, f_{i2} | d_{i-1}, r_{i-1}) = P(d_i | d_{i-1}) P(r_i | r_{i-1}) P(j_i | d_i, r_i, d_{i-1}, r_{i-1}) \cdot P(s_{i1} | j_i) P(f_{i1} | s_{i1}) P(s_{i2} | j_i) P(f_{i2} | s_{i2}) \quad (1)$$

This equation holds for any assignment of values to the random variables involved in the equation. To be more precise, we should have written $d_i = k$, to stand for the

statement variable d_i receives its K 's value, namely, the deformation at the i -th joint is of level k . But we leave the value implicit for notational convenience.

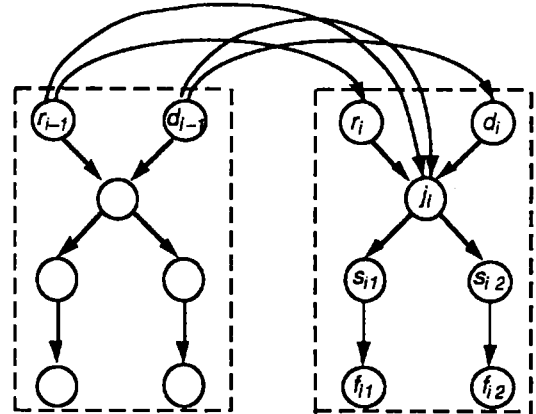


Figure 2: The Bayesian Network of the Assembly $\{A_1, A_2, A_3\}$

Let $d = \{d_i\}, r = \{r_i\}, j = \{j_i\}, s = \{s_{i1}, s_{i2}\}, f = \{f_{i1}, f_{i2}\}$ be sets of random variables formed by running the index i . The joint distribution function represented by Figure 2 is given by:

$$P(d, r, j, s, f) = P(d_0) P(r_0) \prod_{i>0} P(d_i, r_i, j_i, s_{i1}, s_{i2}, f_{i1}, f_{i2} | d_{i-1}, r_{i-1}) \quad (2)$$

This equation forms a joint distribution over all variables of interest from a collection of local distributions listed in the right hand side of Eq. 1. Each of these local distributions can be quite easily quantified. For example, $P(s_{i1} | j_i)$ specifies how disassembly method #1 is effective in decomposing joint i for different hardness levels given by the value of j_i . Similarly, $P(f_{i1} | s_{i1})$ tells us the probability that sensor #1 indicates success given the operation succeeds (e.g., 95%) and given it fails (e.g., 2%). In general, a *Bayesian network* for a set of random variables x_1, \dots, x_n [6] is a directed acyclic graph where each node is associated with a discrete random variable x_i and a local probability table $P(x_i | \pi_i)$, where π_i are the nodes that point into x_i (parents of x_i) and where

$$P(x_1, \dots, x_n) = \prod_i P(x_i | \pi_i) \quad (3)$$

Clearly, not every joint distribution can be represented by a given network because some independence constraints are imposed by Eq. 3. In particular, each node is independent given its parents from all the preceding variables. There are more independence assumptions encoded in the graph, and these assumptions allow efficient computation of posterior probabilities to be made [3]. Inference algorithms for performing these computations are given in [5,6].

Returning to our example of the assembly, we face the need for the following computation. Given that disassembly method #1 failed to decompose joint i , what is the probability that the other method would succeed, that is, compute $P(s_{i2} = 1 | s_{i1} = 0)$. In the language of Bayesian networks, the failure of s_{i1} is called evidence. As evidence accumulates, we proceed by computing new posterior probabilities for decomposition tasks in other parts of the product based on the experience (evidence) accumulated from parts that were subjected to a disassembly operation. Moreover, sensing is also modeled. If a sensor is reliable, then our knowledge of the

success of a certain decomposition operation is sharper and thus influences more drastically our future decisions.

3. The Dynamic Model

The static model described in the previous section serves as a procedure within our decision system. However, it does not take into account the cost of various decomposition methods, the benefits expected from a decomposition or the effects of a decomposition on the product once an action has taken place. We shall now describe a reactive model that performs the needed computations by using the static model for computations of posterior success probabilities and updating its parameters when the result of a disassembly action has been executed.

For modeling the reactive disassembly processes we propose to represent all feasible disassembly sequences in a *Recovery Graph (RG)*, which is a variant of an *And/Or* graph [7,2]. *And/Or* graphs are directed graphs, where edges emanating from the same node are either in an *AND* relation or an *OR* relation with each other. In our context, each node in the *And/Or* graph of a product represents a possible sub-assembly. Edges in the graph emanating from the same node are partitioned via an *AND* relation, so that edges $\{(u,v_0), (u,v_1), \dots, (u,v_m)\}$ are all in *AND* relation to each other if and only if sub-assembly u can be disassembled by a single operation into sub-assemblies v_0, v_1, \dots, v_m . (Equivalently, a single joint connects them to form u). An implicit *OR* relation exists between different *AND* groups emanating from the same node, meaning that if $\{(u,v), (u,g)\}$ and $\{(u,h), (u,x)\}$ are two such groups, then it is possible to disassemble u into either v and g or h and x (see Fig. 3).

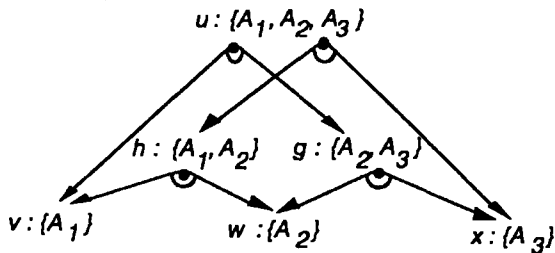


Figure 3: The Recovery Graph of the Assembly $\{A_1, A_2, A_3\}$

It is evident that such an *And/Or* graph is acyclic and that each disassembly plan of the product corresponds to a sub-tree of this graph. The recovery graph of a product is its *And/Or* graph, where with each node and each group of *AND* edges we associate a recovery value. For a node v this recovery value $c(v)$ is the *End-Of-Life (EOL)* value (cost/benefit) incurred by reusing, using-on, utilizing, dumping, or shredding v without further disassembly, that is,

$$c(v) = \text{Max} \{c_{reuse}(v), c_{use_on}(v), c_{utilize}(v), c_{dump}(v), c_{shred}(v)\} \quad (4)$$

For a group of *AND* edges, say $\{(u,v), (u,g)\}$, the recovery cost, $c(u,v,g)$ is the cost of disassembling the sub-assembly, represented by u into the sub-assemblies represented by v and g . To find the *EOL* value of a product and its optimal recovery plan, we associate, in addition with each node v in the recovery graph, an *EOL* value $d(v)$. First, our aim is to calculate $d(p)$, where p is now the node representing the entire product (the root of the graph). We do this by incrementally calculating the $d(v)$ values for all nodes in the graph. To simplify our presentation we assume each *AND* group contains exactly two edges. The generalization to groups with arbitrary number of edges is obvious. For a node v which represents an atomic component $d(v) = c(v)$, since v

cannot be further disassembled. Let u be a node such that for all its sons $d(v)$ is already calculated. Clearly, it would be advantageous to disassemble u into its sub-assemblies, v and g , if and only if:

$c(u) < d(v) + d(g) - c(u,v,g)$. In the same manner, u should not be disassembled any further if for all groups of *AND* sons v, g of u : $c(u) > d(v) + d(g) - c(u,v,g)$. If it should be disassembled, the optimal disassembly operation would be to the *AND* group of nodes which maximizes the expression $d(v_i) + d(g_i) - c(u,v_i,g_i)$ over all *AND* groups v_i, g_i of sons of u . Hence,

$$d(u) = \text{Max} \{c(u), \text{Max}_i \{d(v_i) + d(g_i) - c(u,v_i,g_i)\}\} \quad (5)$$

The computation just described does not take into account the probability of success of a disassembly method. Let $p(s_i | \text{evidence})$ be the probability of success to decompose u using some specific disassembly method. Evidence stands for the sequence of decomposition attempts that led to u or direct observation of other factors. We apply a disassembly action if the expected utility of taking the action is greater than not taking it, that is, if

$$c(u) < p(s_i | \text{evidence})(d(v) + d(g) - c(u,v,g)) + (1 - p(s_i | \text{evidence}))(c(u) - c(u,v,g)) \quad (6)$$

If there are several possible decompositions of u , then we choose a pair v,g that maximizes the right hand side of Eq. 6. To make a disassembly plan we unroll all our possible actions until we reach the atomic components or until an additional decomposition is not worthy due to Eq. 6. Once a decomposition action has been attempted, we update the information using the static model and create a new planning phase, based on the current state and the new information.

The algorithm for the planning phase is best described recursively as follows:

We call a procedure $Eval(p)$ with the root p as its parameter. This procedure returns the *EOL* value of p , as follows:

1. If u is a leaf (has no children), then $Eval(u) = c(u)$. Let $\{g_i, v_i\}$ be u 's children where $\{g_i, v_i\}$ are decompositions of u .
2. Compute the probability of success p_i for disassembling u into $\{g_i, v_i\}$
3. Find the maximum overall decomposition of u :

$$M_i = \text{Max}_i \{p_i [Eval(v_i) + Eval(g_i) - c(u,v_i,g_i)] + (1 - p_i) [c(u) - c(u,v_i,g_i)]\}$$
 (Note that this is a recursive call to $Eval$ with sub-assemblies g_i and v_i).
4. If $c(u) > M_i$ then $Eval(u) = c(u)$;
5. Otherwise $Eval(u) = M_i$.

The reactive mechanism, which takes into account previous decomposition of sub-assemblies is used in the overall decomposition strategy as follows: Once we make a step towards the most promising decomposition, as indicated by the given planning algorithm, that is, when a specific (usually) irreversible action has been taken, we initialize the appropriate node in the static model to true or false, depending on the success of the operation. If sensing is modeled, then the node for the appropriate sensor is instantiated accordingly. The Bayesian network procedures allows us now to compute the success probability of all other attempts based on the successes and failures encountered so far and based on other evidence such as missing parts. Furthermore, costs and benefits that depend on the evidence are updated as well.

This methodology is most suitable for massive disassembly tasks, since it uses the principle of maximum expected utility without taking financial risk factors into account. If a choice should be made between

a value of \$100 given to a manufacturer for a product as is or between investing \$20 in a disassembly attempt with a 50% success rate to obtain a net value of \$120, then clearly the expected utility of the two options is equal. However, the first option is less risky. If the gain would have been slightly larger than \$120 then, as the number of products to be decomposed n increases the risk reduces. If θ_s is taken to be the probability of success of a disassembly method then it is straightforward to compute the probability of a loss as n increases and make sure the loss probability falls below a desired threshold.

4. A Case Study

The disassembly planning approach has been implemented using the HUGIN software [4]. In this section, we present a case study for disassembling a sub-assembly of an obsolete radio which weighs 6 N. Due to space limitations, the model is oversimplified, however, the mechanism of the planning system is demonstrated. The sub-assembly consists of four components: an ABS plastic cover A_1 , an Aluminum case A_2 , a Printed Circuit Board A_3 , and a Transformer A_4 (Fig.4). Let d_1 and d_2 denote the deformation level of A_1 and A_2 respectively, j_i the 'hardness' degree of joint J_i , b_1 the cracking condition level of part A_1 , h_3 the shape of the screws' heads which connect part A_3 , and r_4 the rust of A_4 , and suppose for simplicity that each of these variables has two states. If additional factors have to be modeled, then a larger network would be required and, furthermore, each variable can be described with more states depending on the level of details that is required. The direction of the graph's arcs usually reflect cause and effect relations among the different model's variables. Figure 5 shows a screen of HUGIN that includes the static model. The conditional probability tables can be seen by clicking on each node of the Bayesian Network. For example, $P(s_{11} | j_1 = 0) = 0.95$ and $P(s_{11} | j_1 = 1) = 0.10$ which reflects a success of a non-destructive disassembly method (unscrewing). On the other hand, $P(s_{12} | j_1 = 0) = 0.95$ and $P(s_{12} | j_1 = 1) = 0.95$ which shows that s_{12} is almost always successful since, say, it is a destructive method. Note that when decomposition method s_{11} is sensed as failure, then the success probability of decomposing joint J_2 is decreased, and similarly, the probability of deformation and rust in the rest of the sub-assembly is increased. When a success of a destructive decomposition method s_{12} , which is hardly influenced by rust and deformation, is sensed, then the success probability of decomposing of joint J_2 is hardly affected.

The predictive plan of the product is to decompose the sub-assembly $\{A_1, A_2, A_3, A_4\}$ into $\{A_2, A_3, A_4\}$ and $\{A_1\}$ and furthermore into $\{A_2, A_4\}$ and $\{A_3\}$ and then into $\{A_2\}$ and $\{A_4\}$. This disassembly yields a benefit of \$0.8. Starting with the reactive planning, during the disassembly process we discover that s_{21} , the disassembly operation on $\{A_3\}$, does not succeed, then the updated probabilities imply that it is best to leave the product as is since a failure at the other part $\{A_4\}$ is likely and costly. The resulting sub-assembly is $\{A_2, A_3, A_4\}$ with a benefit of \$0.6.

5. Concluding Remarks

We have introduced a paradigm for planning disassembly processes based on static and dynamic models. The static model is based on the product topology, deterioration processes, and alternative disassembly methods each of which is represented by means of a random variable in a Bayesian Network. The dynamic

model integrates evidence and measurements which are gathered during the disassembly process. This integration creates an appropriate environment for reactive planning, which takes into account the uncertain conditions of the product to be decomposed.

The planning method can be used directly at the disassembly process planning stage as a Process Navigator to assist in the product's dismantling process, or in an autonomous disassembly system. Another promising usage is at the early product design stages, where various possible designs can be modeled and tested from the perspective of their End-Of-Life value.

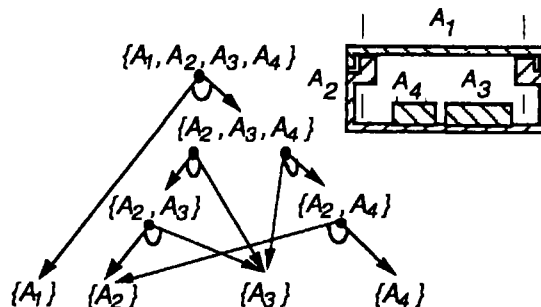


Figure 4: A Sub-Assembly of an Obsolete Radio and its Recovery Graph

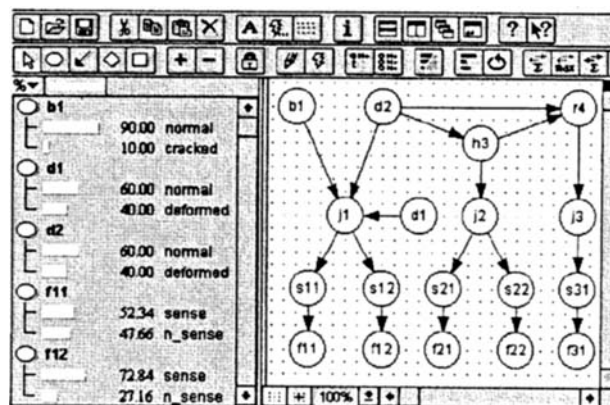


Figure 5: The static model of the sub-assembly

References

- (1) Feldmann, K., Meedt, O., Scheller, H., 1994, Life Cycle Engineering, CIRP - Int. Seminar on Life Cycle Engineering, 195-209, Erlangen, Germany.
- (2) Elmaraghy, W.H., 1993, Evolution and Future Perspectives of CAPP, Annals of the CIRP, 42,2.
- (3) Geiger, D., Verma, T., Pearl, J., 1990, Identifying Independence in Bayes Networks, Networks, 20:507- 34.
- (4) HUGIN - Reference Manual, <http://www.hugin.dk>.
- (5) Lauritzen, S.L., Spiegelhalter, D.J., 1988, Local Computations with Probabilities on Graphical Structures and their Applications to Expert Systems, J. Royal Statist. Soc., B, 50,2:154-227.
- (6) Pearl, J., 1988, Probabilistic Reasoning In Intelligent Systems: Network of Plausible Inference, Morgan Kaufmann.
- (7) Zussman, E., Kriwet, A., Seliger, G., 1994, Disassembly Oriented Assessment Methodology to Support Design for Recycling, Annals of the CIRP 43,1: 9-14.
- (8) Altling, L., Zhang, H.C., 1989, Computer Aided Process Planning: the State of the art Survey, IJPR, 27,4: 553-558.