In: Grastien, A. and Stumptner, M. (eds.),19th International Workshop on Principles of Diagnosis (DX08), Blue Mountains, Australia

Qualitative Modeling for Diagnosis of Machines Transporting Rigid Objects

Peter Struss, Axel Kather, Dominik Schneider, Tobias Voigt

Technische Universität München, Germany

Abstract

We present models of elements of a plant that involves the transportation of lumped material. An application context is provided by a project on diagnosing disturbances in food packaging plants and, more specifically, bottling plants. While there exist models of flow of homogeneous matters, such as liquid material in a hydraulic system, based on simultaneous equations of Kirchhoff/Ohm type, in our project we need to cope with non-negligible transportation time of objects and capture phenomena like the tailback of units (if transportation is blocked) or the propagation of gaps in the flow of units. Because the application context requires compositionality of the model, i.e. local, context-free models of the individual transportation elements, we are also facing the problem that whether or not a single element produces an output flow (or accepts an input flow) cannot be determined solely by the model of this element, but only through modeling the interaction with the subsequent element, which may block the output (or the previous one not providing the input). This issue is addressed by modeling the potential of an existing flow distinctly from the actual occurrence of a flow, an idea which also can enhance models of continuous flow.

1. Introduction

Producing a **library of reusable compositional models** is crucial to the application of model-based diagnosis, but is mainly neglected in the field. This is one of the main obstacles to transfer of the technology into practice. In this paper, we attempt to present an example of a systematic way of producing a model library for an industrial application project, starting from an analysis of application requirements and its impact on appropriate models, the development and validation of a first principles model, and its abstraction into a qualitative diagnosis model. The general task is fault localization in food packaging plants, and the models need to capture the relevant aspects of the flow of objects of different kinds through the plant.

A quick shot at this problem may assume that models describing the flow of some matter in a system, which are quite widespread in model-based systems, e.g. in model-based diagnosis of hydraulic or pneumatic systems or even analog circuits, may do the job. At least under certain simplifying assumptions, mathematical first principles models exist, and it appears to be straightforward to abstract them into adequate input to a model-based problem solver.

Typically, such models assume that the flowing matter is continuous and homogeneous and does not have to be modeled as an object or in its detailed structure. And they usually incorporate the analogies to Kirchhoff's and Ohm's Laws, which leads to simultaneous equations that imply instantaneous propagation of pressure and disregard time needed by the matter to be transported through the system. Our application domain, which is just one instance of a broader class, involves a flow of objects through a plant and, hence, suggests the use of some flow model, but requires dropping some of the simplifying assumptions mentioned. One instance of food packaging plants, which are subject to a diagnosis project we are carrying out, are bottling plants, which we will use as an example in this paper. Such plants involve streams of objects of different types, bottles, crates, and pallets being the most prominent ones.

On the one hand, modeling the transportation of **individual objects** is **prohibitive** or useless. On other hand, the abovementioned flow models of a homogeneous matter fail to capture **essential features**, such as **gaps** in the flow or the creation of a **tailback** by some blockage and its propagation through the plant in **finite time**. Furthermore, an inflow and outflow of a single transportation element of a line cannot definitely be predicted by a local model of this element, because they depend also on the supply of the previous element and the intake capacity of the following one, respectively. As a consequence, we had to develop a model that

- includes transportation times,
- covers interrupted flows,
- handles the exchange of flows between neighboring elements appropriately,

which indicates that models of hydraulic components or analog circuits, which describe flow-like and pressure-like features in steady state, are inadequate for our purpose. The paper focuses on presenting a base model addressing the requirements (section 3), its validation through simulation (section 4), and a qualitative diagnosis model obtained from it (section 5). The diagnosis tool is briefly outlined in section 6. Summary and further challenges are presented in section 7.

Figure 1. *Conveyors of a bottling plant for returnables*

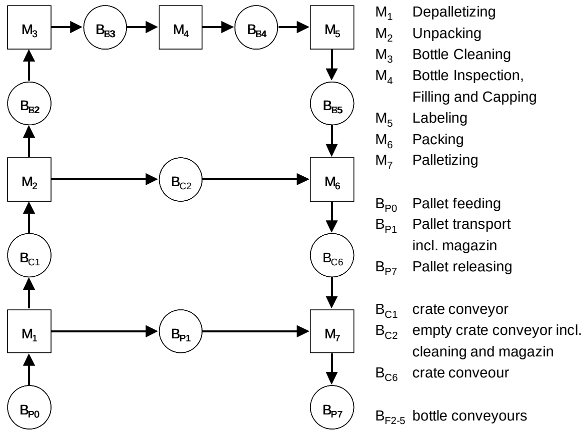

Figure 2. *Generic structure of a bottling plant for returnable bottles*

The following section presents an application context of this work, namely bottling plants

2. An Application Domain: Bottling Plants

Food packaging at industrial scale is carried out in high output packaging lines consisting of specific machines and conveyors. There are different machines for specific packaging tasks, such as primary packaging of food or beverages (e. g. with foil packs, pouches, or containers), secondary packaging (boxes, multipacks, crates, etc.), and tertiary packaging (e. g. pallets or displays). Additionally, machines for de-palletizing and unpacking of returnable bottles, cleaning, inspection and sorting out improper objects may be involved. Plant constellations are configured using one machine of a specific type or several ones in parallel. Machines of different types are connected by conveyors. Because of the high speeds and output rates (up to 100.000 packages per hour), machines and conveyors are failure-sensitive with an availability degree of 92-98 percent. However, as far as downtime of individual machines may have a global impact, the numbers may multiply, causing a much lower availability of the entire plant.
As a specific example for packaging plants, our project considers bottling plants for beverages (e.g. the one shown in Fig 1).
In order to fill beverages into returnable bottles, the material flows of pallets, crates, and bottles (plus labels, glue, etc.) need to be coordinated. This leads to complex line configurations comprised of machines that remove crates from pallets and bottles from crates, process, inspect, or sort objects, and package different types of objects (Fig. 2 shows a simplified and abstract, but typical example).
To prevent oxygen intake or microbiological contamination of the beverage, the filling process should not be interrupted. Therefore, transportation by consecutive machines needs to be decoupled. Otherwise, each individual failure would inevitably cause downtime of the entire plant: In particular, this would stop the filling process and decrease the efficiency of the entire production. To prevent this, the conveyors of bottling plants are designed as transporting buffers like the bottle conveyor in Fig. 1 and the abstract one shown in Fig. 3.
Transporting buffers perform two tasks. One is to carry the objects from one machine to the next one. The other is to store objects in order to be able to compensate for a downtime of the upstream machine and to prevent the immediate propagation of a tailback in case of a downtime of the downstream machine. In addition, the machines located upstream and downstream w.r.t. the filling machine work with higher output rates than the filler. This enables full upstream buffers and receptive downstream buffers to compensate for short downtimes of single machines.
These design principles help achieving a continuous operation of the filling machine. However, in practice, they cannot guarantee avoidance of unwanted idle time of the filler, and (unplanned) downtime of the plant can lie in the range of 10-30 percent.
Machine failures of significant duration, gaps caused by a large number of objects being sorted out, stoppages caused by toppled or jammed objects, or just mistakes of the operators result in downtime of the filling machine and decrease the availability of the entire plant. Because of the interlaced flows of the various object types, time offsets, and the large scale of the plants, the reasons for such plant downtimes can be difficult to identify both by the plant operators and in a retrospect analysis. In consequence,

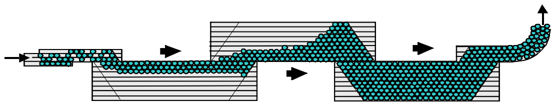

Figure 3. *A three step transporting buffer for bottles*

bottle filling and packaging industries is highly interested in an automated diagnosis tool for their plants that produces information that helps to identify bottlenecks and weaknesses in the plant, related to both the physical performance and configuration and the control principles and parameters. Our tool addresses these needs by localizing the causes for reduced performance of the plant (mainly tied to the throughout of the filler) based on the available recorded data of the machines (over a period of some months or so). In its first version, the system localizes those interruptions of transportation that caused downtime of the filler. (In section 7, we briefly discuss extensions to this scope).

There are a number of requirements and challenges to automated diagnosis raised by this application task. A fundamental economical condition is the fact that many of the potential end users, e.g. breweries, are small or medium enterprises, which could not afford spending many resources on the establishment or adaptation of a tailored diagnosis system for their plant. Another practical requirement is to cheaply accommodate frequent changes in the structure of the line, due to rearrangement or addition of machines. Both issues suggest a **model-based solution** to diagnosis, which allows performing adaptation by simply (re-)specifying the plant structure.

Additional arguments for such a solution stem from the facts that usually a plant is a combination of machines from various manufacturers with different instrumentation and available data and that there may be temporarily missing data due to technical problems. This requires a flexible solution that derives the best diagnosis from **whatever data is available** (in contrast, for instance, to decision trees based on a fixed set of observables).

Heterogeneity and changes of the set of machines also establishes a requirement on the model: firstly, it has to be machine-centered and **compositional**; secondly, it has to be stated at a level of abstraction that covers **types of machines**, independently of specificities and the manufacturer.

Besides these fundamental characteristics, the model has to be capable of properly predicting the propagation of **gaps** in the stream of objects (potentially causing a lack in supply to subsequent machines) and **tailbacks** caused by blockages, as well the **propagation of special features** and deficiencies of the transported objects, which may be caused by improper performance of one machine (e.g. improper cleaning) and may affect the (mis-)behavior of another element downstream (e.g. an inspection machine).

The available **data** is inherently **incomplete** and **imprecise**. Even balance equations do not necessarily hold, because bottles may have been removed by an operator (for inspection or because they blocked the flow) or simply have fallen off the belt. Especially, data (and predictions) about delays and durations involved in the propagation of effects across the various machines are imprecise, since they depend on the state of the machine (e.g. number and distribution of objects in a buffer), acceleration of belts, friction, and other complex behavior aspects.

3. Models of Transportation Elements

3.1 Previous Work

The only similar work we are aware of (except for discrete-event-simulation models used for validation of the control, which do not lend themselves easily to model-based diagnosis) is in the domain of transport of paper in a copier. [Gupta-Struss 95] presents a process-oriented model, and [Fromherz et al. 03] develop a component-oriented model for control generation. Both models are compositional, but focus on the motion of individual sheets, rather than the more abstract perspective of flow of objects.

3.2 Modeling Assumptions

We first list the most important assumptions underlying the transportation models presented here, which are fulfilled in our project domain (under normal conditions), but should also apply to a much broader class of problems.

- The transported objects are rigid bodies with fixed spatial extensions and are not significantly deformed through transportation.
- They are transported with a fixed orientation (like crates), or the orientation does not affect transportation times significantly (e.g. due to a symmetric cross-section, as for bottles).
- There is no interaction among the objects or between objects and the components that has a significant impact on the transportation process (such as bouncing).
- Objects can move only in the direction of the motion of the transportation means (or not at all), although not necessarily with the same speed.

3.3 A Model of a Transportation Element with Buffer

In order to present the essentials of the modeling approach, we consider some sort of archetype of model, which can be specialized or extended to accommodate other kinds of machines. This is a machine that

- has one input and one output with v_{in}, v_{out} being the respective speeds of the means for transportation (e.g. belts),
- possibly transforms or modifies one kind of object (as, for instance, cleaning of bottles), but does not amalgamate several objects to form a new one,
- has a buffer with a (constant) capacity C.

The process of buffering the objects can be fairly random, as illustrated by the bottle conveyor in Figure 3, where bottles may gather in bulks. However, it is assumed, that (under normal behavior) no object is prevented from approaching the output unless it is blocked by other objects ahead, waiting for output. For instance, within the bottle conveyor, its shape and several parallel belts with different speeds ensure that bottles are not left in some corner, but pushed towards the "ideal" fastest belt, if there is space.

Transportation Element with Buffer

State variables

$B(t)$ # objects in buffer

$B_{out}(t)$ # objects buffered for immediate output

$v_{in}(t)$ velocity of input transportation means

$v_{out}(t)$ velocity of output transportation means

$t_d(t)$ minimal transportation time

Parameters

d_0 diameter of transported object (in transportation plain)

C Capacity (as number of objects)

Interface variables

$in.q_{pot}(t)$ potential inflow [objects/s]

$out.q_{pot}(t)$ potential outflow [objects/s]

$in.q_{act}(t)$ actual inflow [objects/s]

$out.q_{act}(t)$ actual outflow [objects/s]

Equations

(1) $in.q_{pot}(t) = v_{in}(t) / d_0$ if $B(t)<C$

$in.q_{pot}(t) = \min (v_{in}(t) /d_0, out.q_{act}(t))$ if $B(t)=C$

(2) $dB/dt = in.q_{act}(t) - out.q_{act}(t)$

(3) $out.q_{pot}(t)= v_{out}(t) /d_0$ if $B_{out}(t)\geq 1$

$out.q_{pot}(t)= \min (in.q_{act}(t - t_d) , v_{out}(t) /d_0)$ else

(4) $dB_{out}(t) /dt = in.q_{act}(t - t_d) - out.q_{act}(t)$

Connector between Transportation Elements

Interface variables

$TE_{n+1}.in.q_{pot}(t)$ potential inflow of upstream element TE_{n+1}

$TE_n.out.q_{pot}(t)$ potential outflow of downstream element TE_n

$TE_{n+1}.in.q_{act}(t)$ actual inflow of upstream element TE_{n+1}

$TE_n.out.q_{act}(t)$ actual outflow of downstream element TE_n

Equations

(5) $TE_n.out.q_{act}(t) = \min (TE_{n+1}.in.q_{pot}(t) , TE_n.out.q_{pot}(t))$

$TE_n.out.q_{act}(t) = TE_{n+1}.in.q_{act}(t)$

Figure 4. *Equations of buffer and connector*

The intuition behind the model can be best described in terms of three fundamental concepts and five "behavior rules", each of which is first introduced informally and then turned into equations. As stated before, one of the problems to be solved stems from the fact that a local machine model in isolation cannot determine whether an actual flow occurs at its input and output. But it can and has to express the limits on the machine's **potential** to take in or output objects. This is reflected by

Concept 1 The **potential input and output flow**, $in.q_{pot}$ and $out.q_{pot}$, represent the maximal flow the machine can accept or generate, dependent on its internal state.

The **actual** flows are represented by two different variables, $in.q_{act}$ and $out.q_{act}$. The first restriction is determined by

Rule 1 The **potential input flow** is given by the input speed of the transportation element, unless the buffer is full. In this case, it cannot be higher than the actual output flow.

In the mathematical model (see Fig. 4), this rule is formalized by equation 1, where d denotes the diameter of the object cross-section and B is the filling degree of the buffer (in terms of number of objects). It involves the assumption that an actual outflow generates the potential for intake instantaneously, which is not true in practice and, hence, another reason for expressing tolerance intervals with values and time. Note that we take all speeds and flows as positive, as their sign is determined by their association with the intrinsic direction of the transportation element. Computing B is straightforward:

Rule 2 The **change in the total** number of buffered objects **is determined by the actual input and output flows.**

The respective equation 2 indicates that B will be computed by integrating the difference of the actual flows. Setting up the model fragments for the potential output flow is based on the second key idea:

Concept 2 B_{out} denotes the number of **buffered output objects** at time t, i.e. the number of objects that can possibly be subject to output at this time.

Before we clarify this crucial concept, we use its intuitive understanding and the third concept for formulating the rule for the potential output flow.

Concept 3 The **minimal transportation time**, t_d, is the time an object needs to get directly from the input to the output, i.e. if it is not delayed by other objects that are piling up.

In case of the bottle conveyor, this means that the bottle stays on the fastest (innermost) belt.

Rule 3 The **potential output flow** is determined solely by the output speed, if there is more than one buffered output object. Otherwise, it cannot be higher than the actual input flow at the time reduced by the minimal transportation time.

One should be aware that in the second case, each single object may (potentially) leave the output with the speed v_{out}. However, if the input flow at the time when it entered was lower, there will be a gap occurring after the output of the object, which makes the (average) flow lower than v_{out}. As a special case, the potential output flow becomes zero, if the actual input flow was zero at the respective time. Again, the respective equation 3 in Figure 4 formalizes this. Computing B_{out} also involves the minimal transportation time t_d. If an object entered the transportation element later than time $t - t_d$, it cannot possibly reach the output at time t and, hence, cannot become part of the buffered output objects. If it entered earlier, it may or may not have already left the output before t, depended on how the actual output flow reduced B_{out}. This consideration is captured by

Rule 4 The change in the **number of buffered output objects** at time t is determined by the actual input flow at time $t - t_d$ diminished by the actual outflow at time t.

Hence, also B_{out} is obtained by integration according to equation 4, which completes the model of the transportation element with buffer. Note, that B_{out} is **not** necessarily the number of objects that form a contiguous pile in front of the output. It could be less, because the last objects that joined the pile entered later than t - t_d.

3.4 Interaction of Transportation Elements

What remains to be done is determining the actual flows from the potential flows of connected machines. This interaction is captured by a model of a generic connector used for connecting all types of transportation elements. The respective rule and equation 5 (Fig. 4) are straightforward:

Rule 5 The **actual output flow of a machine** is limited by both its own potential output flow and the potential input flow of the following machine (and equal to the actual input flow of this machine).

3.5 Other Features and Transportation Elements

The buffer model leaves options for different use and specialization. Due to lack of space, we can only sketch some important cases, many of which are fairly straightforward. For instance, v_{in} and v_{out} could be different as for the entire bottle conveyor shown in Figure 3. In this case, the minimal transportation time t_d needs to be calculated or estimated based on varying speeds along the "ideal path". Alternatively, the same conveyor can be considered as an aggregation of several buffers in series each with one unique speed on its fastest belt, which eases the computation of t_d. Note that the speeds are subject to control and may vary dynamically. Therefore, in case of a unique speed, t_d is determined by the equation

$$l = \int_{t-t_d}^{t} v(\tau) d\tau ,$$

where l is the length of the "ideal path" and v(t) its time-varying speed.

Gates may sit at the input or output of transportation elements and are controlled in a binary manner in order to block the flow entirely if necessary. This is captured by multiplying the respective speed with a factor of (1 – $state_{gate}$), if $state_{gate}$ is 1 for a closed gate and 0 otherwise.

While the bottle conveyor has no fixed relation between the speed of the belts and the motion of the bottles, which may slide, other machines, such as the filler, transport objects by locking them to certain sockets. This is obtained as a specialization of the buffer model with a unique speed and the capacity given by the number of sockets that can be occupied by objects while processing them.

Some elements, such as the bottle cleaning unit, may have n inputs of the same type of objects). To accommodate this feature in the model, we simply have to replace the actual input flow by the sum of several individual input flows. Elements having several outputs (for objects of the same type) usually require some modeling of the mechanism that distributes the objects among the various outputs, e.g. evenly (if possible) or according to some criteria. An example for the latter case is given by inspection machines ejecting objects that fail to pass some test.

Another class of machines produces an output by **combining** objects of different kinds, as for instance the packaging of 20 bottles in a crate. The ratio of the number of different objects participating in this combination is usually not arbitrary, but exactly specified. This ratio links the various potential and actual inflows and the outflow, which is then limited by the "slowest" input flow (relative to the ratio of the respective object type).

The counterpart to this very generic combination element is the **separation** element, with unpackers being a subclass, in which the slowest actual outflow of a decomposition result limits the potential inflow of the composite object.

This set of fairly generic model types turns out to cover the variety of machines in a bottling plant and, more generally, also in the food packaging plants that we encountered.

4. Validation of the Base Model

In order to validate the component models described above we implemented them as numerical simulation models in MATLAB/SIMULINK® [MathWorks 08] and compared the simulated behavior (using the solver \ode4" (Runge-Kutta) with a fixed-step size of one second) with the one of real plants.

Every component was modeled using the equations introduced above and tested in isolation to check whether it was adequate of and stated in a context-independent manner, which is a prerequisite for compositionality. In a second step, a model of a complete plant was configured using the validated components.

In testing the individual components, values of single parameters and variables were varied, and the response of the simulated behavior was monitored. For example, the predicted changes in the buffered material B of a component for different values of the input speed v_{in} and the output speed v_{out} are shown in Figure 5. It depicts that the buffer fills as long as the input speed is higher than the output speed (assuming a sufficient supply), whereas with the input speed reduced to its minimum 0.1 and the output speed being still high, the amount of buffered objects decreases.

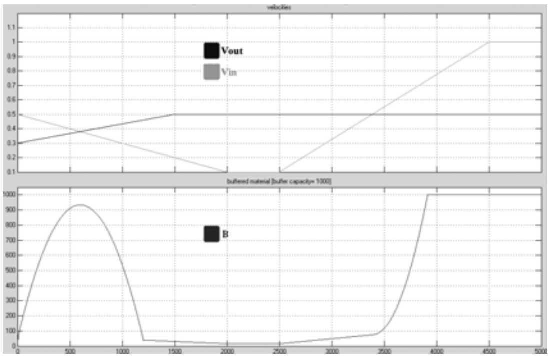

Figure 5. *Plots showing the changes of the buffer (lower) in response to variation of input and output speeds (upper)*

Because of the minimal transportation time, t_d, of the component, the buffer is not completely emptied, as long as there is input available. Furthermore, only the objects represented by the variable B_{out} determine the existence of an output flow. Another real characteristic behavior can be reproduced when increasing the input speed while maintaining the output speed constant. Although v_{in} is still higher than v_{out}, the buffer filling degree remains constant after a certain time, because it is limited by the maximum capacity of the component.

Similar results were achieved by testing the other component type models, providing evidence that the models capture the features relevant to the diagnostic task and do not violate context-independence.

The second challenge was validation by comparing the simulated behavior of a plant model with the behavior of a real plant. Several test cases were constructed, based on real-world downtimes scenarios of the bottling plant whose topology is shown in Fig. 6.

The simulated plant consists of a primary flow of bottles and a secondary object flow of crates. In one test case, the downtime propagation of a failure of the crate washer was simulated and analyzed. This failure interrupts both object flows. After some delay, missing input occurs at the crate packer. Also the unpacker stops at some point, due to its output being blocked. The details of the propagation of failure depend on the capacities and filling degrees of the various buffers connecting the machines. For instance, if the crate magazine is empty and all other buffers are filled with a sufficient degree, the lack of crates will rapidly reach the crate packer. This causes a blockage of the labeling machine and the bottle filler (because the packer is not able to process the bottles) before the lack of bottles in the primary flow (caused by the inoperable unpacker) reaches the filling machine. In contrast, if the crate magazine is completely full, the crate packer keeps working for some time, and the filling machine will be stopped due to a lack of bottles.

Even for this complex scenario, the simulation model reproduces the behavior of the real world plant. Similarly, the characteristics of fault propagation occurring in real plants were predicted for other relevant scenarios.

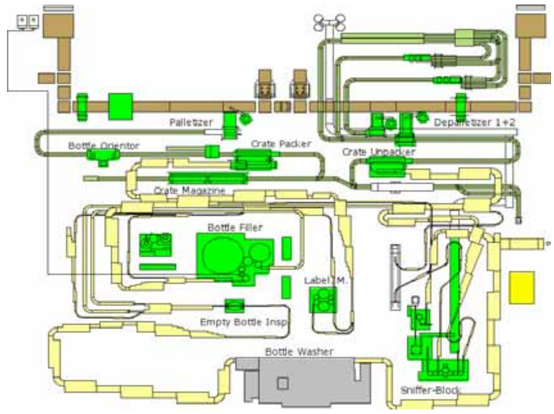

Figure 6. *The structure of one of the test plants*

5. Abstraction to Diagnosis Models

Using the model presented above directly for diagnosis is not appropriate. Firstly, as for all numerical models, its accuracy is only a pretended one in many respects, e.g. in assuming conservation laws to hold and in ignoring the imprecision in the available data, e.g. when flows are determined via counters or the speed of belts. Secondly, the diagnostic task requires the analysis of qualitative, rather than arbitrarily small numerical deviations from the nominal behavior and, hence, needs to be addressed by an appropriate level of abstraction in the model.

The level of model abstraction depends on the intended goal of the diagnosis: we first focused on "hard" failures (stop of the filling machine, that is) caused by hard faults (blockage of another machine), which can be based on distinguishing zero from non-zero flow only. For capturing "soft" faults (deviating behaviors) that lead, perhaps in combination, to a hard failure or a non-optimal behavior, a different model is required.

The total interruption of the flow requires distinctions between zero and non-zero flows only. Sign abstraction of the numerical model yields the qualitative constraints on the variables shown in Fig. 7 (we omit equations (2) and (4), which are difficult or impossible to exploit because neither B(t) nor $B_{out}(t)$ can be observed properly) together with the respective finite relations. (Remember that flows and speeds cannot be negative).

The abstraction of combination elements (such as the crate packer) outlined in section 3.5 will include the application of the three model fragments of Fig. 7 to all individual inflows as well as a constraint simply stating the qualitative equality of all inflows (the ratio of the flows drops out, because it is a positive number):

$$[in_1.q_{pot}(t)] = [in_2.q_{pot}(t)] = \dots = [in_k.q_{pot}(t)].$$

This captures, for instance, the fact that one lacking input will stop all other inputs, as well. The dual applies to separation elements.

Transportation Element with Buffer

(1) $[in.q_{pot}(t)] = [v_{in}(t)]$ if $C\text{-}B(t) > 0$

$[in.q_{pot}(t)] = \min([v_{in}(t)], [out.q_{act}(t)])$ if $C\text{-}B(t) = 0$

$[in.q_{pot}(t)]$	$[v_{in}(t)]$	$[out.q_{act}(t)]$	$[C\text{-}B(t)]$
0	0	*	+
+	+	*	+
+	+	+	0
0	0	+	0
0	+	0	0

(3) $[out.q_{pot}(t)] = [v_{out}(t)]$ if $B_{out}(t)\text{-}1 \geq 0$

$[out.q_{pot}(t)] = \min([in.q_{act}(t - t_d)], [v_{out}(t)])$ if $B_{out}(t)\text{-}1 < 0$

$[out.q_{pot}(t)]$	$[v_{out}(t)]$	$[in.q_{act}(t - t_d)]$	$[B_{out}(t)\text{-}1]$
0	0	*	0
0	0	*	+
+	+	*	0
+	+	*	+
0	0	+	-
0	+	0	-
+	+	+	-

Connector between Transportation Elements

(5) $[TE_n.out.q_{act}(t)] = \min([TE_{n+1}.in.q_{pot}(t)], [TE_n.out.q_{pot}(t)])$

$[TE_n.out.q_{act}(t)] = [TE_{n+1}.in.q_{act}(t)]$

$[TE_n.out.q_{act}(t)]$	$[TE_{n+1}.in.q_{pot}(t)]$	$[TE_n.out.q_{pot}(t)]$
0	0	+
0	+	0
+	+	+

Figure 7. *Sign-based qualitative models of buffer and connector. [x] denotes the sign of x. "*" in a row represents "no restriction" and, hence, the entire row multiple tuples*

We briefly demonstrate that the inferential power of the model suffices for handling the considered class of faults and failures despite its simplicity: assume that a transportation element TE_n with a single speed, $v_{in}(t) = v_{out}(t)$, produces an output, i.e. $[TE_n.out.q_{act}(t)] = +$, but has no inflow, $[TE_n.in.q_{act}(t)] = 0$. Then the constraints yield:

$$[TE_n.out.q_{act}(t)] = + \overset{(5)}{\Rightarrow} [TE_n.out.q_{pot}(t)] = +$$

$$\overset{(3)}{\Rightarrow} [TE_n.v_{out}(t)] = [TE_n.v_{in}(t)] = +$$

$$[TE_n.out.q_{act}(t)] = + \wedge [TE_n.v_{in}(t)] = +$$

$$\overset{(1)}{\Rightarrow} [TE_n.in.q_{pot}(t)] = +$$

$$[TE_n.in.q_{pot}(t)] = + \wedge [TE_n.in.q_{act}(t)] = 0$$

$$\overset{(5)}{\Rightarrow} [TE_{n-1}.out.q_{pot}(t)] = 0$$

If TE_{n-1} is operational, which implies $[TE_{n-1}.v_{out}(t)] = +$, then

$$[TE_{n-1}.out.q_{pot}(t)] = 0 \wedge [TE_{n-1}.v_{out}(t)] = +$$

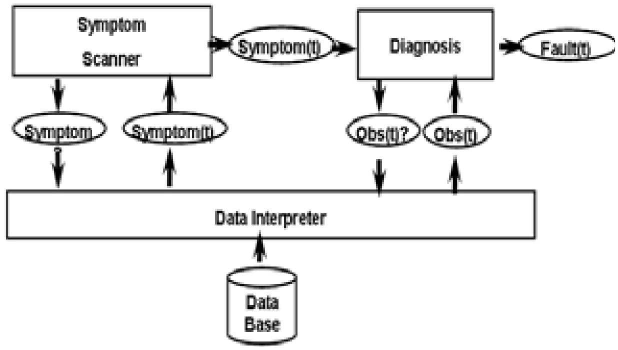

Figure 8. *The architecture of the tool*

$$\overset{(3)}{\Rightarrow} [TE_{n-1}.in.q_{act}(t - t_d)] = 0 .$$

This means, even without information about the buffers, the lack is propagated backwards across the models of correct elements (but will be consistent with a "blocked" mode, for instance) as expected.

The base model is also a starting point for the generation of a diagnostic **deviation model**, which captures the propagation of disturbances that do not lead to a complete interruption of the flow, but still to a suboptimal performance of the plant (mainly indicated by a reduced speed of the filler). It is described in [Struss et al. 2008].

6. Use of the Model in the Diagnosis Tool

In the diagnostic model, the delay t_d has to be replaced by an **interval** that reflects the approximate nature of the model and uncertain or lacking data (e.g. about the buffer and speeds of transportation elements). For consistency-based diagnosis, these intervals have to be chosen in a very **conservative** way to avoid spurious inconsistencies. As a result, repeated prediction would blow up the temporal uncertainty tremendously. Therefore, the diagnosis algorithm interleaves prediction and observation, narrowing down the predicted time interval to the observed one whenever a prediction is confirmed by the data.

The architecture of the diagnostic analysis tool is shown in Fig. 8. The data base is scanned for **initial symptoms**. In the current version, this is the stop of the filler. But symptoms can be specified also as reduced speed of the filler or more complex behavior patterns. The Data Interpreter establishes a link between the higher level concepts of symptoms and the model (tailback, lack, modes of components, etc.) on the one hand and the recorded data on the other hand. It will return detected symptoms with an associated time interval, and each occurrence will trigger diagnosis.

The model-based Diagnosis Module generates queries for the various model-based predictions associated with some time interval, which are then confirmed, refuted, or left unanswered in case there is no evidence in the data base. If confirmed, the predicted (conservative) time interval will be replaced by the (usually smaller) observed duration, which is then used for further prediction. The consistency-

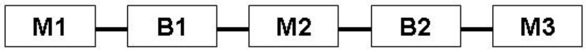

Figure 9. *Three machines M_i with buffers B_i in between*

based diagnosis engine will then generate diagnostic hypotheses, which represent machine faults associated with some temporal extension. Each **diagnostic hypothesis** comes with **evidence** in the form of a **set of positive or negative observations** extracted from the data base that helps the experts to comprehend and further analyze the situation.

So far, the model and the diagnosis tool have been evaluated on data collected for two plants and properly localized the reasons for filler stops.

7. Summary and Outlook

The model and the overall solution we described have been developed based on a thorough analysis of the application and its requirements – not only under technical, but also organizational and economic aspects. Cheap generation and adaptation of a solution for a new or modified plant is a crucial requirement and a major argument in favor of a model-based solution. However, the precondition for such a solution is the creation of models that capture the essential behavioral features, but are still generic and provide a library from which a plant model can be generated without additional work on modeling specific features of individual machines.

The validation of the model through simulation and its use in analyzing real data from two different plants has provided evidence that the models really capture the essential features of plant behavior we are interested in from a diagnostic perspective.

Also, the adaptation of the tool to the available data in each plant has to be easy and cheap. In our architecture, the Data Interpreter decouples the high-level, generic representation of the behavior features from the specificities of the structure and content of the recorded data.

Furthermore, the project aims at a contribution to improving the general conditions through standardization of the data acquisition. Partners of the consortium are the originators of an existing standard that has now been widely accepted for bottling plants. This has been extended on the one hand regarding data relevant to diagnosis and on the other hand generalizing it for food packaging plants. This will significantly improve the conditions for effective and easily adaptable diagnostic solutions.

There still remains a problem that challenges the common view on component-oriented diagnosis.

7.1 Another Challenge: Multiple Culprits

The diagnostic task described so far, namely identifying the cause of filler stop or slowdown as a fault in one of the other machines (or bad input, such as too many improper bottles), appears to be straightforward. In reality, there are cases in which such an answer is at least questionable or even misleading. This has to do with the fact that the plant achieves some tolerance with respect to limited disturbances of individual machines through buffer elements.

To illustrate this, we consider a sequence of three machines M_i with two intermediate buffers B_j. If buffer B_2 is filled to some extent, it will prevent that some short interruption of output from M_2 will cause missing input to M_3. Now assume that M_1 experienced a series of several small disturbances over a longer period of time ("stuttering"), none of which caused a lack of input to M_2 or M_3. However, as an accumulated effect of this stuttering, the amount of objects buffered in B_2 has been reduced significantly, affecting its capability to compensate for further interruption of flow. As a result, a fairly short fault in M_2, which would have had no serious effect under "normal" circumstances, now causes a lack at M_3, and the algorithm described would actually detect this and blame solely M_2 for the stop of M_3.

In order to properly understand the origin of the symptom and for determining appropriate remedies, it would be necessary to reveal the role of M_1 as the originator or, at least, one contributor to the problem. While it is possible to extend the time windows and the observed behavior patterns in the tool, is not obvious whether and how to distribute the blame among the components, even more if there are more complex situations of disturbances in several machines.

The core of this problem lies in the fact that there is an element of the designed plant behavior that is not captured by the local component models: although there is no global control, there do exist some principles and intentions how to run the plant properly. For instance, as mentioned in section 2, the output rates of the machines should increase both upstream and downstream from the filler (to avoid filler lacks and tailbacks, respectively). For the same reason, the filling degree of buffers should be "neither too high nor too low". Unfortunately, this healthy degree is not fixed and, more importantly cannot be measured directly or appropriately estimated from other sensor data.

Handling such kind of situations properly will be a challenge to the remaining work in this project.

References

[Fromherz et al. 03] Fromherz, M., Bobrow, D., and de Kleer, J.: Model-based Computing for Design and Control of Reconfigurable Systems. In: Qualitative Reasoning. Special issue of AI Magazine, Winter 2003, 24(4), AAAI Press, Menlo Park, USA

[Gupta-Struss 95] Gupta, V., Struss, P.: Modeling a Copier Paper Path: A Case Study in Modeling Transportation Processes. In: QR-95, Working Papers of the Ninth International Workshop on Qualitative Reasoning, Amsterdam, 1995.

[MathWorks 08] http://www.mathworks.com/products

[Struss 04] P. Struss: Models of Behavior Deviations in Model-based Systems. In: Proceeding of the 16th European Conference on Artificial Intelligence (ECAI04), 2004

[Struss et al. 08] Struss, P., Kather, A., Schneider, D., and Voigt, T.: Qualitative Modeling for Diagnosis of Machines Transporting Rigid Objects. In: Bradley, L. and Trave-Massuyes, L (eds.). 22nd International Workshop on Qualitative Reasoning QR08, Boulder, Co, 2008