Modeling a Copier Paper Path: A Case Study in Modeling Transportation Processes

Vineet Gupta* Peter Struss†

TR-95-019

May 1995

Abstract

We present a compositional model of paper transportation in a photocopier that is meant to support different problem solving tasks like simulation and diagnosis, and to be applicable to a variety of configurations. Therefore, we try to avoid making hard-wired implicit assumptions about design principles and possible scenarios. In order to simplify our analysis, the model abstracts away from the physical forces and reasons only about velocities. Nonetheless, it succeeds in determining essential features of the motion of the sheet of paper like buckling and tearing. The framework provided is quite generic and can be used as a starting point for developing models of other transportation domains.

*Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto CA 94304 USA. (vgupta@parc.xerox.com)

†Technical University of Munich, Orleansstr. 34, D-81667 Munich, Germany. (struss@informatik.tu-muenchen.de)

1 Introduction

A photocopier is a device with a clear and fixed structure that seems to lend itself to standard component-based modeling techniques. However, as already pointed out in reports about earlier modeling attempts [S+87], this is true only if we ignore the sheets of paper that move around and interact with the copier's components, often in a way that forces us as users to sequences of recovery actions. Paper handling in a copier is an instance of a class of transportation processes whose modeling requires the modeling of the transported subject as well.

An attempt to model the forces, motions and deformation of a sheet of paper under the influence of some rollers can lead to quite sophisticated mathematical models [SL81b, SL81a]. As a result, the work we report here had to address two basic problems:

- modeling a device with a topology changing over time.
- finding an appropriate level of abstraction of modeling that still allows us to determine most of the interesting features, for instance buckling or tearing of paper sheets.

Furthermore, guiding principles were to

- exploit features of the physical domain for simplifying the model, but avoid hardwired implicit assumptions about possible scenarios, design principles etc.
- aim at reusable model fragments that can be further specialized and composed to cover different instances of devices, and tasks such as simulation, design and diagnosis.

The key step was to create locally acting transportation processes (in our domain the interaction between a sheet and, for instance, a pair of rollers). and of the interaction between them mediated by the transported subject, the sheet. This is a fairly general approach and should be of use in modeling other types of transportation devices as well.

In this paper, we present a description of an intermediate conceptual model which is still under development and implementation. It succeeds in determining essential features of the motion of the sheet, despite its simplicity that stems from considering *velocities only*.

We briefly introduce the domain by discussing an example of a copier paper path (section 2) and present the intuition underlying the model in section 3. Section 4 describes the model formally and attempts to reveal the underlying assumptions and restrictions. We apply the model to an example in section 5, and summarize in section 6.

2 The Domain: Paper Transportation in a Photocopier

Our problem arose in the specific domain of photocopiers, so in this section we describe the paper path of a simple photocopier, and we will illustrate the concepts developed in this paper on this machine.

In this photocopier, paper is loaded in a paper tray at the left of the machine (see Fig. 1). When a signal is received, the acquisition roll is lowered onto the paper and pulls the top sheet of paper towards the first set of rollers (1). (We will assume here that there is some mechanism to ensure that it pulls only one sheet.) After the paper is grasped by the first set of rollers, the acquisition roll is lifted, and the rollers pull the paper forward. The image is transferred onto the paper by the image transfer

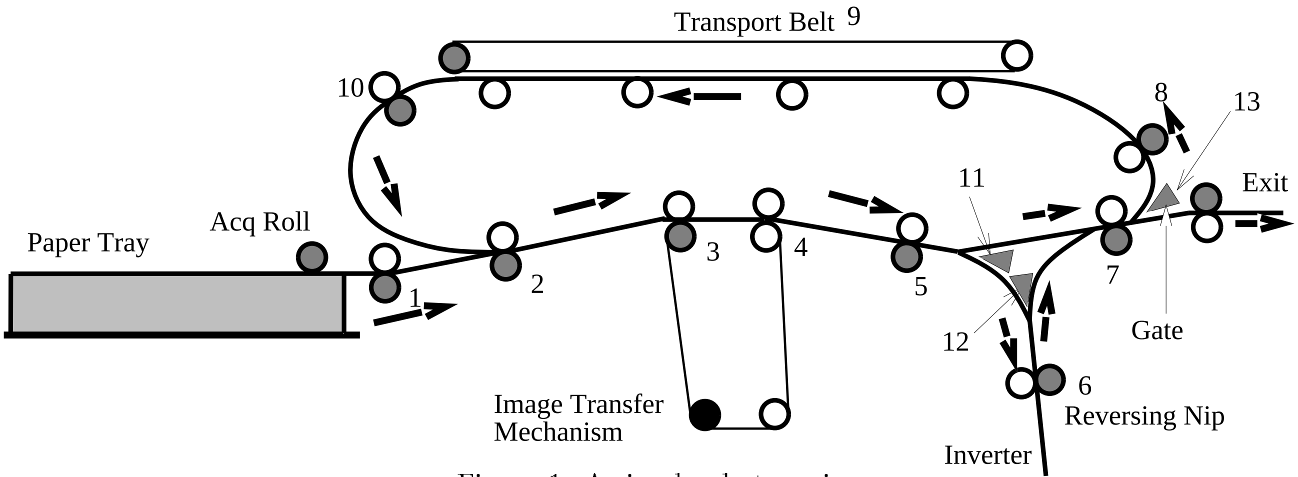

Figure 1: A simple photocopier

mechanism, and it then travels to the right. If it is desired that the paper be printed on both sides, it is inverted by diverting it into the inverter by means of a gate. The inverter rolls pull it in, then stop, and push it out to the right. It then goes through the top loop, where it is stopped at roller number 10, thus slipping on belt 9. On receiving another signal, roller 10 pulls it forward, and then the other side is printed. Then it is inverted again and sent through the exit into a tray.

This simple copier has most of the kinds of transportation elements that are found in the real machines — rollers, reversible rollers, belts etc. Most of the interesting situations in a real copier are illustrated here. The sheet can be under more than one driven roller (rollers number 1 and 2), the rollers can change directions (6), and there is a belt (9), which has low sliding friction. In addition, rollers have different speeds, for example we will assume that the belt and roller 8 in the upper loop of the path move faster than the rollers in the lower part of the path. Thus the paper is pulled by the roller pair number 8.

[SL81b] have shown that under normal operating conditions the coefficient of friction between paper and rollers is sufficiently high to prevent any slipping. On the other hand, the friction between belts and paper is quite low, so sufficient slipping may occur. We will assume that the paper itself has negligible inertia. Hence when it comes into contact with a roller, it is instantaneously accelerated, and when it is in contact with the belt, the small friction force is sufficiently large to move the paper at the same speed as the belt (unless some other transportation element prevents this).

There are of course several scenarios which no copier would probably contain, however our model is sufficiently general to consider these. For example, if we had two successive rollers moving in opposite directions, then the paper would tear. If the back roller moves faster than the front roller, the paper can buckle. Our model is able to predict this. This is necessary from several points of view. Firstly the designer may have made an error, in which case simulating with this model would reveal the error. Secondly something may have gone wrong in the machine, then this prediction would be required for diagnosis. And in any case this is necessary from general design principles, since we want our model to work with as few assumptions as possible, and assuming good design is certainly a strong assumption.

3 Modeling the Paper Transportation — The Intuition

Before we present the formal model of the paper transportation, we try to convey its intuitive background. The overall motion of the transported sheet, as well as potential deformations of the sheet,

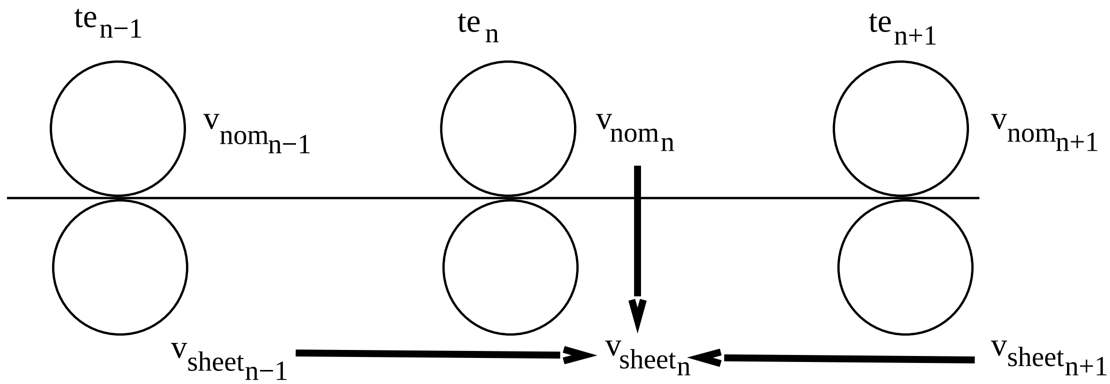

Figure 2: The influences on the local actual speed of a sheet

such as buckling or tearing, result from interaction of the sheet with one or more belts, pairs of rollers, or other potential means (or its inertia, if there is no such interaction). Each interaction of this kind is local, and it is influenced by the type and actual properties of two basic classes of objects: the *sheet*, which can be a transparency, paper, etc. and accordingly have different properties (smoothness of the surface, stiffness, resistance against tearing etc.), and the *transportation elements* of different kinds: for instance, a roller with a clutch may spin faster than with its nominal speed if driven by a sheet, whereas a belt does not. In combination, these objects determine, in particular, whether or not slipping occurs, i.e. the speed of the sheet at this point deviates from the speed of the transportation element. However, this local interaction does not necessarily determine the actual speed of the sheet at this location completely. This also depends on a potential impact of other transportation elements acting on the sheet. For instance, a faster front roller may make the sheet's speed exceed the nominal speed of the second one. Or a nip can completely stop the sheet, thus letting it slide on the rotating belt.

A model that actually attempts to explicitly capture all forces acting on and within the sheet and the way they combine their effects becomes quite complicated, as it would lead to considerations of acceleration, mass, inertia etc. Our solution that a) avoids to explicitly introduce the forces and b) keeps the analysis local, is based on the following idea: If the nominal speed of a transportation element te_n (see Fig. 2) does not differ from the actual speed of the sheet at the immediate neighboring transportation elements, te_{n-1}, te_{n+1} (if extant), then there is no reason for the actual speed of the paper at location n to deviate from the nominal velocity of te_n. If a neighboring speed of the paper is different from this, it may cause a different speed of the paper at location n. Whether or not this actually occurs, depends on the property and current status of the segment of the sheet between the neighboring transportation elements.

For instance, if the sheet is straight between te_n and te_{n+1} and $v_{sheet_{n+1}}$ is greater than v_{nom_n}, this will tend to accelerate the sheet at location n, or tear the sheet. If the sheet was buckled between n and $n+1$, this "pulling forward" would not occur. Similarly, the sheet could be "pushed" at location n, if and only if the segment between $n-1$ and n is stiff (i.e. not, or not very, buckled).

Thus the actual speed of the sheet is influenced by three quantities: the local nominal speed and (potentially) two neighboring actual speeds (Fig. 2). Note, that by considering the neighboring actual speed, the analysis can be kept local. These actual speeds may well differ form the respective nominal ones, as a result of an impact futher downstream, say, an even faster te_{n+2}.

We can get more than merely the statement about the kind of influences occurring, namely *bounds on the local speed of the sheet*: the paper may travel faster or more slowly compared to the nominal speed of the transportation element (by means of the clutch or through slipping). But if it does, it cannot go faster (or more slowly, resp.) than both neighboring speeds. This is true if we assume that nothing else accelerates the paper at this location (e.g. internal elastic forces). As a result, we obtain a chain of local constraints which have to be satisfied, thus determining (or restricting) the local speed of the sheet and actual properties of the segments, such as buckling or tearing.

We will now turn this intuitive understanding into a set of model fragments.

4 Description of the Model

4.1 The Paper Path

In principle, a sheet transported in a copier corresponds to a 2-dimensional surface being deformed and moving around in a 3-dimensional space in a continuous way (unless it tears). A model based on this spatial representation would be very complex. It turns out that a simpler representation suffices to solve many interesting problems in the domain. It is obtained by exploiting the fact that there exists a 2-dimensional surface that is (supposed to be) the set of possible locations of a sheet. This is the *paperpath* whose projection is shown in Fig. 1. Of course, restricting the potential locations of a sheet entirely to the paperpath corresponds to the built-in assumption that the sheet actually travels along the intended path. This prevents the model from properly handling fault situations and cases of bad design.

The model we present here includes an even stronger simplification by considering a *cross-section* of the actual 2-dimensional paper path. We mention that this decision actually excludes the explicit representation of some phenomena relevant to the copier domain, e.g. skewing of the paper due to different speeds of the left and right rollers. Another 2-dimensional feature which occurs even under normal operation is the buckling of the sheet in a direction orthogonal to the cross-section, created in order to increase the stiffness of the sheet, i.e. prevent buckling in the direction of the paperpath.

The cross-sectional representation of the paperpath as for instance depicted in Fig. 1 is a set of contiguous curve sections. We ignore the curvature of these sections and consider distances only along the trajectory[1] This gives us locally a representation by an interval of the real number line—except at branch points, which we have to take into account as they occur in our domain as illustrated by our introductory example. Indexing the maximal linear sections of the paperpath by integers $i \in \mathbb{N}$, and denoting the length of the section by $length_i$, we represent a section by a pair $section(i) = (i, (0, length_i)) \in \mathbb{N} \times I(\mathbb{R}_0^+)$, where $I(\mathbb{R}_0^+)$ is the set of intervals on the positive reals (including 0). Let PATHSECTIONS be the set of such pairs. The point locations on the sections are then described by SECTIONPOINTS $= \{(i, l_i) \mid 0 < l_i < length_i\} \subset \mathbb{N} \times \mathbb{R}$. The structure of the path is captured by a set of *branching points* that sit between sections and specify which sections are possibly connected (perhaps depending on the state of a gate). Let BRANCHINGPOINTS be the set of these points.

Thus the paperpath is represented by

$$\text{PATH} = \text{SECTIONPOINTS} \cup \text{BRANCHINGPOINTS}$$

with an appropriate topology on PATH that treats the branching points as least upper bounds or greatest lower bound resp. of the section intervals. Thus each branching point b_k determines a set of

[1]This is another simplifying assumption — in many analyses it is necessary to consider curvature as it determines the weight of the sheet. Also the path cannot curve too sharply.

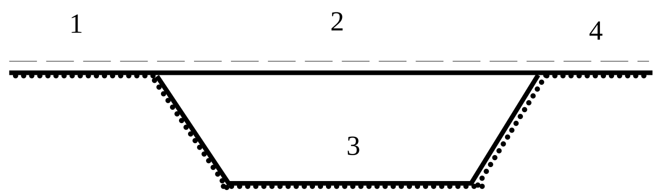

Figure 3: Two different intervals on PATH (dashed line and dotted line) with the same endpoints.

(potentially) connected sections, i.e. all sections that intersect with arbitrarily small open neighborhoods of b_k. Let $connections(b_k) = \{(i,j) \mid i \neq j\} \subset \mathbb{N} \times \mathbb{N}$ denote the set of (theoretically) possible section transitions of b_k. As sheets cover a contiguous part of the path (by definition, if they are torn apart they are considered to constitute several sheets), we need to define *intervals* of PATH. Note that because of branching, it is not enough to specify the lower and upper bound of an interval. This may lead to ambiguity as indicated by the scenario in Fig. 3.

We define an interval I_p of PATH as an image of the interval $[0,1] \subset \mathbb{R}$ under a continuous bijective mapping. Thus possible positions of sheets on a paper path are intervals. We denote by I(PATH) the set of intervals on PATH. The *length* $|I_p|$ of I_p is a real number, and is simply the sum of the lengths of the component sections and tail fragments. The *lower bound* and *upper bound* can be defined as the images of the points 0 and 1. Note that due to branching, intersection of intervals is not necessarily an interval.

In order to simplify the presentation, we will assume that the mapping of $[0,1]$ to PATH (later on representing a sheet or a segment thereof) uniquely induces the same orientation on all sections in I_p. This is possible as we assume that the sheet cannot cover the same section twice (by being folded backward).

To describe the motion of points along PATH, we need a generalized *derivative* of PATHPOINTS that takes transitions at branching points into account: for $p \in PATH$,

$$\frac{d}{dt}p = (trans, v) \in (\mathbb{N} \times \mathbb{N}) \times \mathbb{R}$$

, where v is the real-valued velocity, and $trans = (i,j)$ specifies a transition from one section to another one. Of course, this is interesting only at branching points, as for any point on section i, $trans = (i,i)$. $trans$ has to be continuous at branching points in the sense that it can change from (i,i) to (j,j) only by going through (i,j). Note that $trans = (i,i)$ is possible at a branch point, but only if an extremum is reached and $v = 0$. If $v \neq 0$, then the topology enforces $trans \in connections(b_k)$.

With this representation of the paperpath the problem of modeling the motion of a sheet is basically turned into a 1-dimensional locally linear problem which can be handled by real-valued variables, except for branching points, where a decision has to be made about the continuation of motion.

4.2 Transportation Elements

We now turn to the models of the objects and processes invloved in the paper transportation. For this purpose, we use a representation inspired by the CML proposal [F+94] and close to QPE [For84] which readers may be more familiar with (we do not claim however, that what we present here are valid CML models!).

Each concept class will be described by entries for the following slots:

- *Subclass-of*: specifying links for inheriting entries from other concepts.

- *Parameters*: listing name and domain of static properties, e.g. $length \in \mathbb{R}$ for a sheet.
- *Variables*: listing name and domain of features that may dynamically change over time, e.g. $pos\_int \in I(PATH)$ for the position interval of a sheet.
- *Participants*: specifying instances of concepts that have to be present, and their names (used as local variables), e.g. te : TransportationElement is one participant in the SheetTransportation process.
- *Conditions*: statements that have to be satisfied to establish the existence or the activity of the model, e.g. $te.engaged = T$ is necessary for a SheetTransportation, where *engaged* is a variable in the participant *te*.
- *Consequences*: Statements that are implied by the existence or activity of the model, such as $|pos\_int| \leq length$ for a sheet.

We will first describe the components of the copier that drive the sheets along the paper path — the respective models are shown in Table 1. Any *transportation element* has a parameter *surfacetype* which will be used to distinguish between different effects of friction e.g. between a belt which allows slipping of the sheet (as in belt 9) and a roller that does not. Furthermore, a transportation element has a position *pos_int* which is an interval on the paperpath. If we wish to idealize the interaction between a transportation element and the sheet as happening at *a single point* this can be realized by making *pos_int* a one-point interval. Note that we introduce *pos_int* as a parameter, which means it is considered to be fixed. Thus we do not cover transportation elements that may act at various positions or themselves move while transporting the sheet. It does not imply, however, that the component is always physically located at *pos_int*. The variable denotes the interval on the path where it is supposed to interact with the sheet, provided it is engaged. It may be displaced, for instance due to disengagement or a fault. *engaged* is the variable that specifies this.

We assume that transportation elements do not overlap on the paper path, and that there is no branching happening within *pos_int*. The variable v_{nom} denotes the *speed* with which the component would carry the paper along the interval *pos_int* in the absence of all other forces. It is assumed to be constant along *pos_int*, and is determined by the speed of the motor that drives it. Although for most components v_{nom} is fixed, there are exceptions: the inverter roller (roller 6 in Fig. 1) changes the direction of its movement, and roller 10 goes from 0 to a positive velocity. Finally, there is the actual speed of the transportation element v_{act}, which may differ from the nominal speed. Different types of components impose different relationships. This is captured by the different subclasses of TransportationElement in Table 1. There are rollers with a clutch that allows the roller to spin faster than the motor driving it, namely if the sheet is pulled by a faster roller upstream. In contrast, a transportation element without a clutch enforces equality of v_{nom} and v_{act}. Copier belts are examples of such components.

Note that these models cover a variety of different physical mechanisms, that are used in such components, for instance normal spring or vacuum forces. If these details matter, e.g. for diagnostics, the components classes could be further specialized.

4.3 Sheet and Sheet Segments

The second basic class of objects is, of course, the *sheets* that are being moved around. Parameters characterizing a sheet are its *length*, a *surfacetype*, that together with that of the transportation element

Transportation Element
    Parameters
        $pos\_int \in I(PATH)$
        $surfacetype \in STYPES$
    Variables
        $engaged \in \{T, F\}$
        $v_{nom}, v_{act} \in \mathbb{R}$

RollerWithClutch
    Subclass-of
        TransportationElement
    Consequences
        $\lvert v_{act}\rvert \geq \lvert v_{nom}\rvert$

TEWithoutClutch
    Subclass-of
        TransportationElement
    Consequences
        $v_{act} = v_{nom}$

Belt
    Subclass-of
        TEWithoutClutch

Table 1: Model fragments for transportation elements.

determines sliding phenomena, and *stiffness*, which allows us to determine whether or not a sheet will buckle (Table 2). Our current model treats these properties as homogeneous across the whole sheet. Future extensions have to drop this restriction — the reader is familiar with transparencies with a white strip attached for increasing friction.

A sheet has a position *pos_int* on the path, i.e. occupies an interval on PATH. For reasons of compact representation we introduce sheet conditions *straight* and *buckled*, which give the relation between the length of the sheet and the length of *pos_int*, as shown in the consequences of Sheet in Table 2.

We add another variable, *tearing*, a flag that indicates that the sheet is being torn apart. This should last only for an instant: as soon as a sheet is torn, we stop the simulation, and can raise an exception. The fact that a sheet is not torn is captured by the consequence $|pos\_int| \leq length$.

As already indicated in Fig. 2, we consider a sheet with n transportation elements acting on it, where $n > 0$. Since transportation elements are assumed to have disjoint positions on the path, this induces a segmentation of the sheet — an alternating sequence of segments on which the transportation elements act or not (Fig. 4). Our model treats these segments as the primitive "parts" of a sheet that together determine the motion and condition of a sheet through their own motions and conditions.

Sheet

Parameters

- $length \in \mathbb{R}$
- $surfacetype \in STYPES$
- $stiffness \in \mathbb{R}$

Variables

- $pos\_int \in I(PATH)$
- $condition \in \{straight, buckled\}$
- $tearing \in \{T, F\}$

Consequences

- $|pos\_int| \leq length$
- $condition = straight \Leftrightarrow |pos\_int| = length$
- $condition = buckled \Leftrightarrow |pos\_int| < length$

SheetSegment

Participants

- sht: Sheet

Variables

- $pos\_int \in I(PATH)$
- $length \in \mathbb{R}$
- $v_{left}, v_{right} \in \mathbb{R}$
- $impact_{left}, impact_{right} \in \mathbb{R}$
- $trans_{left}, trans_{right} \in \mathbb{N} \times \mathbb{N}$
- $condition \in \{straight, buckled\}$
- $stiffness, tearing \in \{T, F\}$

Conditions

- $pos\_int \subseteq sht.pos\_int$

Consequences

- $|pos\_int| \leq length$
- $\frac{d}{dt} lb(pos\_int) = (trans_{left}, v_{left})$
- $\frac{d}{dt} ub(pos\_int) = (trans_{right}, v_{right})$
- $condition = straight \Leftrightarrow |pos\_int| = length$
- $condition = buckled \Leftrightarrow |pos\_int| > length$
- $sht.condition = straight \Rightarrow condition = straight$
- $condition = buckled \Rightarrow sht.condition = buckled$
- $tearing = T \Rightarrow sht.tearing = T$
- $stiffness = T \Rightarrow condition = straight$
- $impact_{left} \times impact_{right} \leq 0$
- $stiffness = f(sht.stiffness, |pos\_int|)$
- $\forall s : \text{SheetSegment}.(s.sht = sht \wedge$ $s.pos\_int \cap pos\_int \neq \emptyset \Rightarrow s.pos\_int = pos\_int)$

ContactSegment

Subclass-of

- SheetSegment

Participants

- st : SheetTransportation

Conditions

- $pos\_int = st.contactint$
- $sht = st.sht$

Consequences

- $v_{left} = v_{right}$
- $condition = straight$

FreeSegment

Subclass-of

- SheetSegment

Conditions

- $lb(pos\_int) = lb(sht.pos\_int) \vee \exists cs : \text{ContactSegment}$ $(sht = cs.sht \wedge ub(cs.pos\_int) = lb(pos\_int))$
- $ub(pos\_int) = ub(sht.pos\_int) \vee \exists cs : \text{ContactSegment}$ $(sht = cs.sht \wedge lb(cs.pos\_int) = ub(pos\_int))$

Consequences

- $condition = buckled \Rightarrow impact_{left} = impact_{right} = 0$
- $condition = straight \wedge v_{right} > v_{left} \Rightarrow tearing = T$
- $\frac{:tearing=F}{tearing=F}$

LeftTailSegment

Subclass-of

- FreeSegment

Participants

- cs : ContactSegment

Variables

- $leading, trailing \in \{T, F\}$
- $trans_{left} \in \mathbb{N} \times \mathbb{N}$

Conditions

- $sht = cs.sht$
- $lb(pos\_int) = lb(sht.pos\_int)$
- $ub(pos\_int) = lb(cs.contactint)$

Consequences

- $impact_{right} = 0$
- $condition = buckled \Rightarrow v_{left} = 0$
- $condition = straight \Rightarrow v_{left} = v_{right}$
- $leading = T \Leftrightarrow v_{left} < 0$
- $trailing = T \Leftrightarrow v_{left} > 0$
- $trailing = T \Rightarrow (trans_{left} = (i, j) \Leftrightarrow$ $\exists pos\_int' \subset pos\_int, |pos\_int'| > 0 \wedge$ $lb(pos\_int') = lb(pos\_int) \wedge pos\_int' \subset section(j))$

InternalSegment

Subclass-of

- FreeSegment

Participants

- cs_{left}, cs_{right} : ContactSegment

Conditions

- $sht = cs_{left}.sht = cs_{right}.sht$
- $lb(pos\_int) = ub(cs_{left}.pos\_int)$
- $ub(pos\_int) = lb(cs_{right}.pos\_int)$

Consequences

- $condition = straight \wedge stiffness = F$ $\Rightarrow impact_{left} = max(0, v_{right} - cs_{left}.st.te.v_{nom})$ $\wedge impact_{right} = min(0, v_{left} - cs_{right}.st.te.v_{nom})$
- $stiffness = T$ $\Rightarrow impact_{left} = v_{right} - cs_{left}.st.te.v_{nom}$ $\wedge impact_{right} = v_{left} - cs_{right}.st.te.v_{nom}$

Table 2: Model fragments for the sheet.

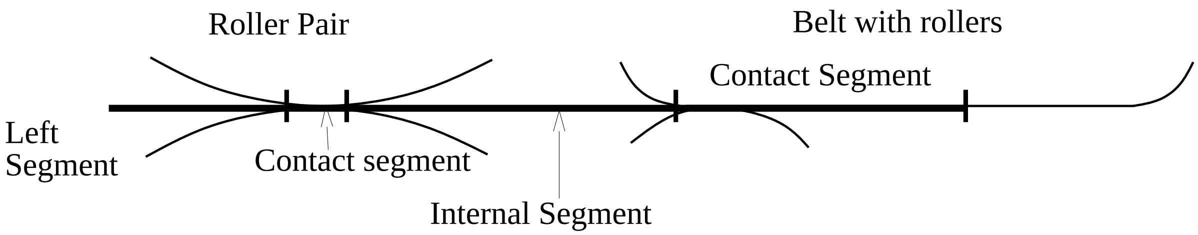

Figure 4: Segments of a sheet.

Note that the segments are "virtual" parts in the sense that, over time, they represent different physical portions of the sheet and that they are created, destroyed and changed in type as the sheet moves along the path.

As captured by the concept of *SheetSegment* (Table 2), each segment belongs to one particular sheet, has a position *pos_int* on the path and a *length*. Also, like a sheet, it can be buckled or straight, with the same definitions, and tearing, and has some stiffness (which may change for a segment depending upon its length). The magnitude of $length - |pos\_int|$ determines the *amount of buckling*, which has to be limited in a copier.

The consequences express certain obvious constraints between the *condition* of a sheet and a segment, for instance, that all segments of a straight sheet are straight, too.

For the purpose of modeling the transportation, the key part concerns velocities which are considered at the left edge and right edge of the segment's position only. Furthermore we introduce variables $impact_{left}$ and $impact_{right}$, that represent the potential impact the forces and motion of the right-hand edge of the segment may have on the left-hand edge and vice versa, as already explained in the previous section. ($impact_{left}$ means impact *on* the left because of the right).

The constraint $impact_{left} \times impact_{right} \leq 0$ in the consequences is a central one. It eliminates implicit cycles in causality: both impacts cannot go in the same direction. Intuitively, if the right side tends to "pull" at the left side, then the left side cannot simultaneously "push" the right. They could however pull each other. Finally we have a consequence stating that no two segments of the same sheet may overlap, this comes from our intuition that the sheet cannot fold back on itself.

As indicated above, we distinguish between different classes of segments depending on how they are related to the transportation elements. *ContactSegments* (Table 2 and Fig. 4) are the segments directly under the influence of a transportation element. This influence will be modeled in the *SheetTransportation* process described in Section 4.4. Here it suffices to understand that a SheetTransportation is related to one sheet and has a position *contactint* that is the overlap of the positions of the sheet and the TransportationElement. The condition for a ContactSegment is, hence, that it shares both with an existing SheetTransportation. Our model assumes that both ends have the same velocities and that no buckling occurs — actually this might be violated on a belt, at least under fault conditions.

The other segments are *FreeSegments*. The two conditions say that FreeSegments are maximal in the sense that they are on both sides limited by the edges of the sheet or the boundary of a contact segment. The first consequence states that a buckled sheet does not transmit any impact from one side to another, an aspect of the simplified binary treatment of buckling. Furthermore, a straight segment is tearing if its right speed exceeds its left speed, but the last consequence makes tearing an exception, reflecting the assumption that sheets are usually strong. It is a default rule in the sense of [Rei80] and means that if the model allows for a solution without tearing, then tearing will remain false. In the

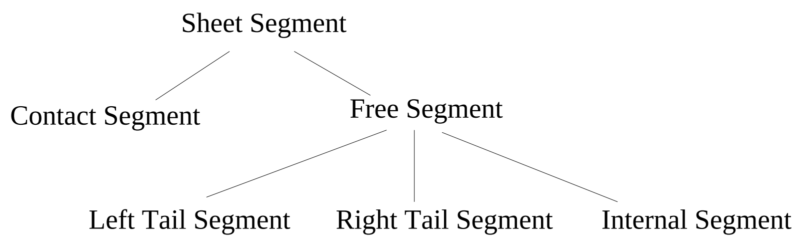

Figure 5: Class hierarchy of sheet segments.

comparison of velocities at different pathpoints, we exploit the assumption stated in Section 4.1 that the sections are aligned.

The first kind of segments are tail segments, which have a contact segment on one side only. The class *LeftTailSegment* describes such a situation. Since nothing acts on the left edge, the impact on the right side, where the transportation element applies, is 0. The *RightTailSegment* is analogous. In the current model we do not consider situations when no transportation element is acting on the sheet (e.g. at the output). This would require considering gravity and inertia, both of which are avoided in our model.

We distinguish between a tail segment *leading* i.e. moving its edge beyond a currently occupied position, or *trailing*. This is relevant as the leading tail segment determines the motion of a sheet, particularly at branching points (see Section 4.5), while the trailing segment just follows the sheet. The latter behavior is expressed by the last consequence in LeftTailSegment. Note that for a buckled sheet both edges could be leading or trailing.

Finally, there are the free segments with contact segments on both sides, the class *Internal Segments*. The core of the model is again the set of constraints on the *impact* variables. For example, a straight segment that is not stiff can only "pull". This means that $impact_{left}$ is always positive and it may increase the speed of the sheet on the left above the nominal speed of the transportation element there up to the speed on the right. Similarly $impact_{right}$ can only be negative, "pulling backwards". However, a stiff segment can "pull" and "push", so its impacts can be in either direction. The last consequence indicates that whether a segment is stiff or not depends on the sheet's stiffness and the length of the segment. This binary valued model will be further refined, and may include a dependency on whether a segment is straight or not. The concept of the impacts captures the intuition we tried to convey in the previous section. Their combined effect is computed by the SheetTransportation.

4.4 Sheet Transportation

This concept (Table 3) involves a transportation element and a sheet. Its conditions state that the transportation occurs if the transportation element is engaged and some part of the sheet is in contact with it. The contact position is given by the intersection of the positions of the sheet and the transportation elements, and may vary as the sheet travels through the transportation element. The next consequence states that there must be a unique contact segment of the sheet corresponding to this sheet transportation, this is called *contseg*, and is used in the later consequences. The *impact* variables receive inputs from the left and right free segments, if they exist. The left and right speeds are also identified with the speeds of any adjacent segments.

The next consequence is the crucial one for determining the actual speed of the contact segment.

SheetTransportation

Participants

sht : Sheet

te : TransportationElement

Variables

$contactint \in I(PATH)$

$slide \in \{T, F\}$

$impact_{left} \in \mathbb{R}$

$impact_{right} \in \mathbb{R}$

Conditions

$te.engaged = T$

$te.pos\_int \cap sht.pos\_int \neq \emptyset$

Consequences

$contactint = te.pos\_int \cap sht.pos\_int$

$\exists! contseg$: ContactSegment$.contseg.sht = sht$
$\wedge contseg.pos\_int = contactint$

If $\exists seg_{left}$: FreeSegment$[sht = seg_{left}.sht$
$\wedge ub(seg_{left}.pos\_int) = lb(contactint)]$
then $impact_{left} = seg_{left}.impact_{right}$
$\wedge v_{left} = seg_{left}.v_{right}$
else $impact_{left} = 0$

If $\exists seg_{right}$: FreeSegment$[sht = seg_{right}.sht$
$\wedge lb(seg_{right}.pos\_int) = ub(contactint)]$
then $impact_{right} = seg_{right}.impact_{left}$
$\wedge v_{right} = seg_{right}.v_{left}$
else $impact_{right} = 0$

$min(te.v_{nom} + impact_{left}, te.v_{nom} + impact_{right})$
$\leq contseg.v_{left}$
$\leq max(te.v_{nom} + impact_{left}, te.v_{nom} + impact_{right})$

$slide = f(te.surfacetype, sht.surfacetype)$
$slide = F \Rightarrow te.v_{act} = contseg.v_{left}$

$min(te.v_{nom}, contseg.v_{left})$
$\leq te.v_{act}$
$\leq max(te.v_{nom}, contseg.v_{left})$

Table 3: Model fragment for SheetTransportation.

The speed is bounded by the potential modifications of the nominal velocity of the transportation element through the impacts of the neighboring free segments. What is actually expressed by the inequality is some explicit resolution of the influences on the speed of the segment, and hence involves a closed world assumption — if there were other forces acting on the paper, it might violate these bounds.

The second contribution of the Sheet Transportation is to determine the actual speed v_{act} of the transportation element. Remember that it has been related to v_{nom} in models for the transportation elements, e.g. by $|v_{act}| \geq |v_{nom}|$ if a clutch is present. If no sliding occurs (determined in a binary way from the surfacetypes), v_{act} equals the segment speed. The last consequence expresses that if v_{act} deviates from v_{nom}, then this can be only due to the segment driving it, and is hence bounded by its speed.

As indicated in the previous section, we now have constrains on the velocities local w.r.t. the SheetTransportation processes. These interact by transmitted impacts through free segments. This is the basis for determining bounds for the speeds of the segment edges and of the entire sheet. In section 5, we illustrate this by applying the model to a non-trivial example.

4.5 Branching

What remains to be modeled is the motion at branching points. In the copier, this is controlled by gates which either permanently (gate 12 in Fig. 1) or dynamically (gates 11 and 13) restrict the section transitions that are possible in principle ($connections(branch)$) to a subset. This is captured by the *Gate* model in Table 4. We have to introduce gates at all branching points, even though the restriction of the section transitions may be due to geometry rather than a real physical gate (e.g. near roller 2 in Fig. 1).

As stated earlier, the direction of the motion is determined by a *leading tail segment*. Hence, the *BranchingLeft* process (Table 4) involves a Gate and a LeftTailSegment, which has to be leading and have its edge at the branching point of the gate. The consequences then restrict $trans_{left}$ of the tail segment to the connections allowed by the gate. Continuity of $trans_{left}$ (see Section 4.1) will select those that match the section from where the edge is approaching. In a designed artifact, this should uniquely determine the transition under normal conditions. Again, *BranchingRight* is symmetric to this model.

Gate
Parameters
$branch$: BRANCHINGPOINTS
Variables
$connections \in \mathbb{N} \times \mathbb{N}$
Conditions
$connections \subset connections(branch)$
BranchingLeft
Participants
$tseg$: LeftTailSegment
$gate$: Gate
Conditions
$tseg.leading = T$
$lb(tseg.pos\_int) = gate.branch$
Consequences
$tseg.trans_{left} \in gate.connections$

Table 4: Model fragment for gates and branching.

5 Example

As an example to illustrate the use of our model, we apply it to the scenario in Fig. 6. Scenarios in copiers are much simpler than this due to design principles.

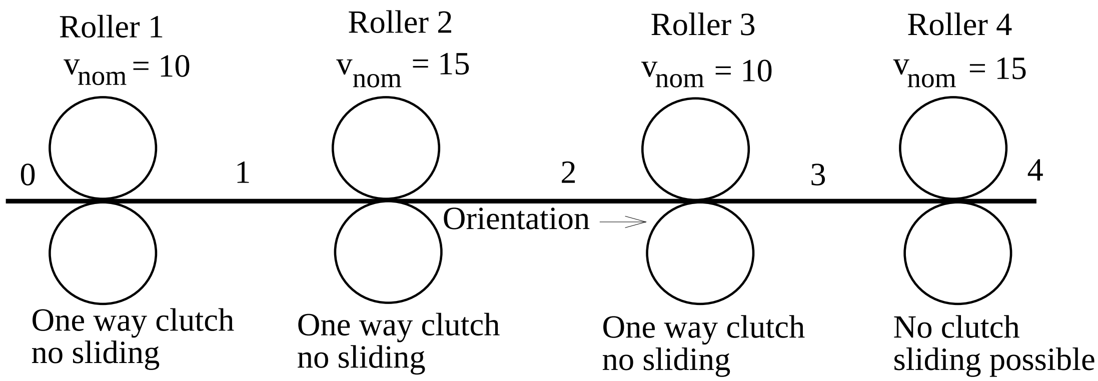

Figure 6: An example scenario.

The sheet is in contact with all 4 rollers. It is initially straight, and is not stiff anywhere. All rollers contact the sheet at a point, so all contact segments are of length 0. We will thus identify the v_{left} and v_{right}, and simply call it v_i for each contact segment i, where a contact segment is given the number of the transportation element it is under.

There are 5 free segments, of which segment 0 is the left tail segment, and segment 4 is the right tail segment, while all the others are internal segments. The sheet is not stiff, so none of the segments is stiff.

As there are 4 transportation elements, there are 4 instances of the class SheetTransportation. For the first 3, we assume slide = F, whereas for the fourth, slide = T.

Thus we get the following equations from the various model fragments.

$$\begin{array}{rcll} Roller1.v_{act} & \geq & 10 & \text{(clutch)} \\ v_1 & = & Roller1.v_{act} & \text{(no sliding)} \\ Roller2.v_{act} & \geq & 15 & \text{(clutch)} \\ v_2 & = & Roller2.v_{act} & \text{(no sliding)} \\ Roller3.v_{act} & \geq & 10 & \text{(clutch)} \\ v_3 & = & Roller3.v_{act} & \text{(no sliding)} \\ Roller4.v_{act} & = & 15 & \text{(no clutch)} \\ Seg(i).v_{left} & = & v_i = Seg(i-1).v_{right} & i = 1,2,3,4 \end{array}$$

Besides this we also get equations for the impacts on either side of the segments. Substituting some values from the equations above, we obtain the following. The second to seventh equations follow as the sheet is straight and not stiff.

$$
\begin{array}{rcll}
Seg0.impact_{right} & = & 0 & \text{(left tail)} \\
Seg1.impact_{left} & = & max(0, v_2 - 10) & \\
Seg1.impact_{right} & = & min(0, v_1 - 15) & \\
Seg2.impact_{left} & = & max(0, v_3 - 15) & \\
Seg2.impact_{right} & = & min(0, v_2 - 10) & \\
Seg3.impact_{left} & = & max(0, v_4 - 10) & \\
Seg3.impact_{right} & = & min(0, v_3 - 15) & \\
Seg4.impact_{left} & = & 0 & \text{(right tail)}
\end{array}
$$

From the model fragment for SheetTransportation 1, we thus get the following inequality.

$$10 \leq v_1 \leq max(10, max(10, v_2))$$

Since $v_2 \geq 15$, this means $10 \leq v_1 \leq v_2$. Thus $Seg1.impact_{left} > 0$, so $Seg1.impact_{right} \leq 0$.

Now we apply this information to the equation for SheetTransportation 2. We then get

$$15 \leq v_2 \leq max(15, max(15, v_3))$$

If $v_3 > 15$, then $Seg2.impact_{left} > 0$. This means that $Seg2.impact_{right} = 0$, thus we would get from the equation for SheetTransportation 3 that $Seg3.impact_{left} > 5$. But that would mean that $Seg3.impact_{right} \leq 0$, which would give $v_4 = 15$, thus $Seg3.impact_{left} = 5$. This is a contradiction, thus $v_3 \leq 15$, so we conclude $v_2 = 15$.

Now we can conclude that $Seg2.impact_{right} = 0$, so the inequalities for SheetTransportation 3 and SheetTransportation 4 yield:

$$
\begin{array}{l}
10 \leq v_3 \leq max(10, v_4) \\
min(15, v_3) \leq v_4 \leq max(15, min(15, v_3))
\end{array}
$$

Note that $v_3 > 15$ gave us a contradiction before. So $v_3 \leq 15$. Thus we get $10 \leq v_3 \leq v_4 \leq 15$.

Since there are no other equations available, we apply the no-tearing default rule to Seg1 and Seg3. This gives us $v_2 \leq v_1$ as $tearing = F$, thus $v_1 = v_2$. Similarly, we get $v_3 = v_4$, which gives us $10 \leq v_3 = v_4 \leq 15$.

There are no more equations, so this is the best result our model can provide. And it agrees with what one would expect: it tells us that the sheet has a speed 15 at rollers 1 and 2, roller 1 spins faster than its nominal speed. At rollers 3 and 4 the speed is between 10 and 15. Thus segments 1 and 3 remain straight, while segment 2 may buckle if $v_3 < 15$, and then sliding would occur at roller 4. A more accurate model which had information on the relative friction forces between the rollers and paper would enable us to give precise numbers for v_3 and v_4, rather than an interval.

6 Summary and Discussion

The work presented here is a first step towards a declarative and general model of paper handling in a copier. The concepts, however, are more general and should be useful in other transportation processes (e.g. in textile manufacturing or printing). The key point is splitting the overall process into locally acting transportation processes, which are then related by impacts transmitted by the transported object. Particular properties, such as elasticity, of these objects can then be expressed at these points.

The model provides a good coverage of the actual and conceivable scenarios in paper handling, with some limitations pointed out in section 4. This has been achieved at a fairly abstract qualitative level of representation that refers to velocities only.

A weakness of the model that needs to be overcome is the binary treatment of features like stiffness, sliding etc. A related deficiency is the necessity to resolve some ambiguities through the non-tearing default. A solution to these problems has to take forces into account, for instance through a qualitative comparison of friction to the strength of the sheet. We are currently working on such an extension and will present it separately. The key idea is to analyze and compare forces in the same structural scheme as the velocities, and to assume that everything is at equilibrium. Thus we can draw free-body diagrams for each of the components, the resulting constraints are solved to determine the motion of the paper. This analysis determines which features occur, e.g. sliding or tearing, and hence the appropriate velocity model. In this way we still avoid a detailed study of acceleration. However, a "gold-standard" model that includes forces, accelerations, mass and inertia, has to be related to our model in order to reveal its assumptions and limitations (e.g. using the theory of model transformation and simplification of [Str92]).

As of now, this model exists and has been analyzed on paper. We plan to formulate it as a domain theory in CML ([F+94]), and also implement it in `tcc` ([SJG94, SJG95]), a reactive declarative constraint-based language. An efficient implementation would have to exploit good solutions to the creation, modification and elimination of sheet segments. This is possible in `tcc` since these actions are tied to particular events — a sheet entering or leaving a transportation element, engagement of a transportation element, tearing etc.

In order to evaluate the model and its implementation, the first goal is to perform simulation of a photocopier, and use it to check the control code. Later we will use it for code generation, explanation and diagnosis also.

Acknowledgements. We gratefully acknowledge the discussions with David Bell, Danny Bobrow, Brian Falkenhainer, Markus Fromherz, Vijay Saraswat and Mark Shirley, and thank them for their valuable comments. Peter would also like to thank the International Computer Science Institute at Berkeley and Xerox PARC for support of his research visit in Berkeley.

References

[F+94] B. Falkenhainer et al. CML: A compositional modeling language. Unpublished manuscript., November 1994.

[For84] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[Rei80] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81 – 132, 1980.

[S+87] Jeff Shrager et al. Issues in the pragmatics of qualitative modeling: Lessons learned from a xerographics project. *Comm. of the ACM*, 30:1036–1047, 1987.

[SJG94] V. A. Saraswat, R. Jagadeesan, and V. Gupta. Foundations of Timed Concurrent Constraint Programming. In Samson Abramsky, editor, *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Press, July 1994.

[SJG95] V. A. Saraswat, R. Jagadeesan, and V. Gupta. Default Timed Concurrent Constraint Programming. In *Proceedings of Twenty Second ACM Symposium on Principles of Programming Languages, San Francisco*, January 1995.

[SL81a] T.C. Soong and C. Li. The rolling contact of two elastic-layer-covered cylinders driving a loaded sheet in the nip. *J. of Applied Mechanics*, 48:889–894, 1981.

[SL81b] T.C. Soong and C. Li. The steady rolling contact of two elastic layer bonded cylinders with a sheet in the nip. *Int J. of Mech. Sci.*, 23:263–273, 1981.

[Str92] Peter Struss. What's in SD? towards a theory of modeling for diagnosis. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-based Diagnosis*. Morgan-Kaufmann, 1992.