

Media transformations for the representation and communication of multimedia production activities

S. J. Morris
City University,
Department of Information Science,
Northampton Square,
London EC1V 0HB, U.K.
Tel: +44 171 433 8344
Fax: +44 171 433 8584
E-mail: sjm@soi.city.ac.uk

Abstract

Case studies made of multimedia document production highlight the need for a means of classifying and describing the transformations of media elements which make up this process. The classification set out here contains twelve types belonging to two categories, constructive and supportive. A set of transformation representation rules provides the framework for succinct communication between production participants. This communication can serve both a descriptive function, describing events as the basis for design rationale, or a prescriptive function outlining detailed stages before they are activated.

Keywords

media transformations, multimedia, production process, communication

1 INTRODUCTION

The work reported here constitutes the results from one section of a project whose primary objective was to provide a 'production method' for multimedia documents akin to the 'design methods' which are already part of software engineering. The general theme of this project was the integration, within one method, of the activities involved in the 'design' of document content and those required for the 'development' of software that provide structure and access mechanisms. The project reached positive conclusions regarding :

a) the usefulness of two new concepts, *navigable discourse structure* and *media transformations*,

b) the possibility of combining these concepts in a new discourse driven model of the production process, which provides a sound foundation for practical guidance, and,

c) the effectiveness of a staged method for the production of multimedia documents which uses this model as a framework.

An overview of the whole project appears elsewhere (Morris and Finkelstein, 1996). This paper concentrates on the concept of media transformations, a new means of describing and prescribing changes to media elements during multimedia document production. Media transformations offer a means of communication between production participants responsible for either activity.

The approach taken in the main project combined two complementary areas of study. One part of the work was more theoretical, the examination of concepts and theories relevant to the design and development of multimedia. The other part was more practical, the experimental production of multimedia documents using the standard multimedia tool, Director (Macromedia, 1994). The intention of this second part was to examine practical problems of production and, in the absence of any systematic studies, to elucidate the production process. It was from this second part that the notion of media transformations arose. The purpose of these experimental productions was to recreate live demonstrations of two software engineering tools, the *viewer* (Nuseibeh and Finkelstein, 1992) and the System Architect's Assistant (Kramer *et al.*, 1993), as freestanding multimedia documents. The *viewer* is a prototype environment supporting the framework known as Viewpoint Oriented Software Engineering (VOSE) (Nuseibeh & Finkelstein, 1992). The first case study produced the *viewerdemo*. The second case study involved the production of the System Architect's Assistant Demonstration (SAAD).

A live demonstration of a software engineering tool, such as the *viewer*, and a computer based recreation of such a demonstration, for instance the *viewerdemo*, illustrate the differences between multimedia and multiple media propounded in the project. Both are composite media objects, with different structures but the same communicative purpose, which is to show what the *viewer* does and why it is important. The live demonstration is an example of multiple media. The demonstration incorporates several abstract media including still and moving images, graphics, natural language speech, computer languages, and text. The carriers for these abstract media include printing on paper, sound waves and the virtual medium provided by the computer. The combination of all these physical and abstract media takes place at the time of the demonstration and at the discretion of the presenter. The multiple media are held together by the actions and utterances of the presenter, who provides co-ordination by the selection and timing of these elements. The *viewerdemo* is an example of multimedia. It uses the same abstract media as its live predecessor, but they are carried together in a physical medium

holding digital signals. This single physical medium is created, stored, manipulated and displayed exclusively in the computer. Integration of the abstract media takes place at the discretion of the developer and the resulting structure determines the manner of presentation to the audience.

Practical experience of producing these multimedia demonstrations highlighted the need for a way of classifying and representing activities. Such a representation would facilitate the planning of production, its subsequent recording and the continuing communication between participants. The crucial factors are the order and nature of the activities involved in the production process.

Knowledge about current and previous document states provides the basis for decisions about the order of operations. In the *viewerdemo* there was an initial listing of the potential sources for media elements and the creation of new versions of these sets of data was recorded, but without any precise indication of the activities that caused changes to come about. The lack of a suitable basis for a 'design rationale' (Carroll and Moran, 1991) was the cause of this deficiency. A general classification of possible multimedia production activities would offer a useful foundation. It would provide both a means of describing existing *ad hoc* methods and a means for defining new and more systematic approaches. The lack of such a means of describing activities seriously inhibited the logging of the production of the *viewerdemo*. In general it also forestalls any attempt to construct any general approach to systematic design.

A preliminary, static and three-level model of production provides an initial context for examining media transformations. Any method must allow development to take place on at least three different levels (Morris and Finkelstein, 1993). These are the levels of discourse structure, media composition and disposition, and presentation, shown diagrammatically in Figure 1. Work may progress sequentially, concurrently or in some set of temporal combinations according to the method proposed. The arrows shown on Figure 1 indicate the possibility of simultaneous instantiation of related elements on different levels. The level of discourse structure is the most abstract. It is here that any underlying communicative purpose of the artifact acquires a coherent form. Choice, disposition and composition of individual media elements take place in the media composition and disposition level. The level of presentation involves both the design of the appearance of the final artefact and the finalisation of its structure as a computer based object.

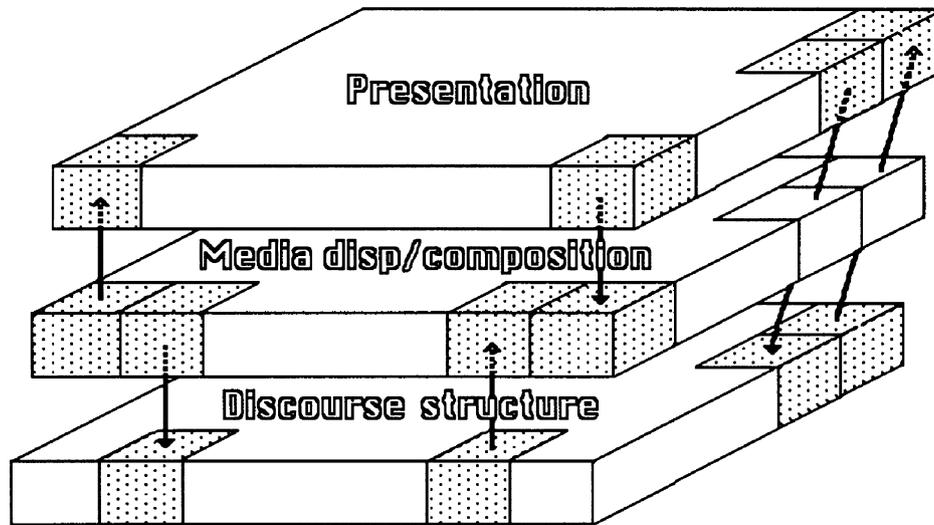


Figure 1 Three levels for multimedia document production.

2 DYNAMIC CHANGE VIA MEDIA TRANSFORMATIONS

Media transformations can introduce an essential notion of dynamic change into the static description of multimedia document production already given. Definition, at the level of media disposition and composition, of the media elements available will define the current state of this part of the document, whether interim or final. The questions left open are how movement between such states takes place and what the process of change involves. Similar questions apply at the presentation level of the static model.

In order to elucidate these problems, the earlier model is now extended to show two new features, the components generated at each level and the relationships of components between levels. A diagrammatic view appears as Figure 2. A discourse structure of a general type appears as an hierarchy with internal compositional relationships (shown as broken-line arrows). The abstract media selected and generated at the next level have an external relationship with the discourse components (shown as double-headed arrows), representing each singly or in some multiple combination. These media elements are then presented, individually or jointly, in a spread carried on the virtual physical medium, or may be attached to it as one of the access operations which are themselves also abstract media elements (relationships shown as arrows).

The process of change between intermediate document states involves a series of media transformations, each drawing on elements from earlier stages to create modified or new elements. Figure 3 illustrates diagrammatically how two sets of transformations, T1 and T2, relate three states, 0, 1 and 2, of a document in production. The unconnected media element in state 1 indicates that design is in an ill-defined or intermediate state requiring attention.

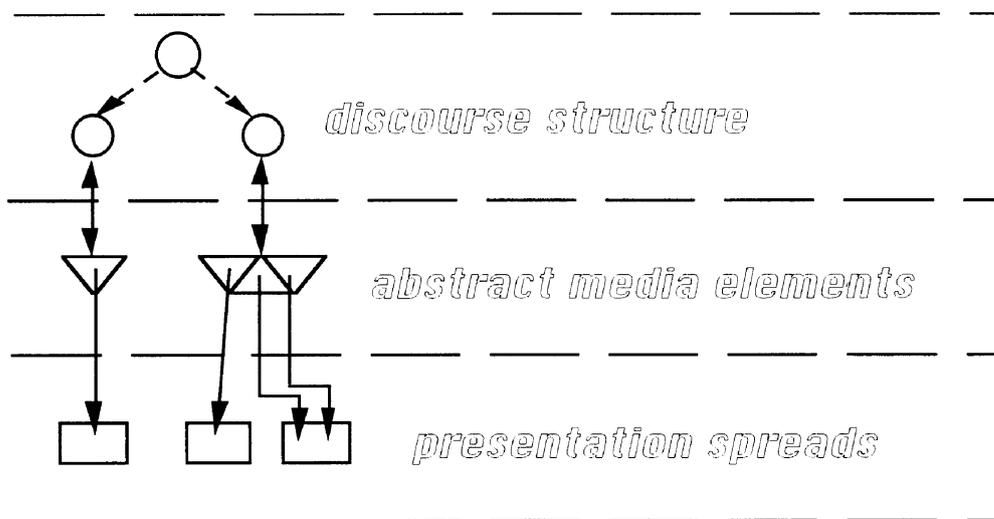


Figure 2 Relationships between levels of earlier production model (Figure 1).

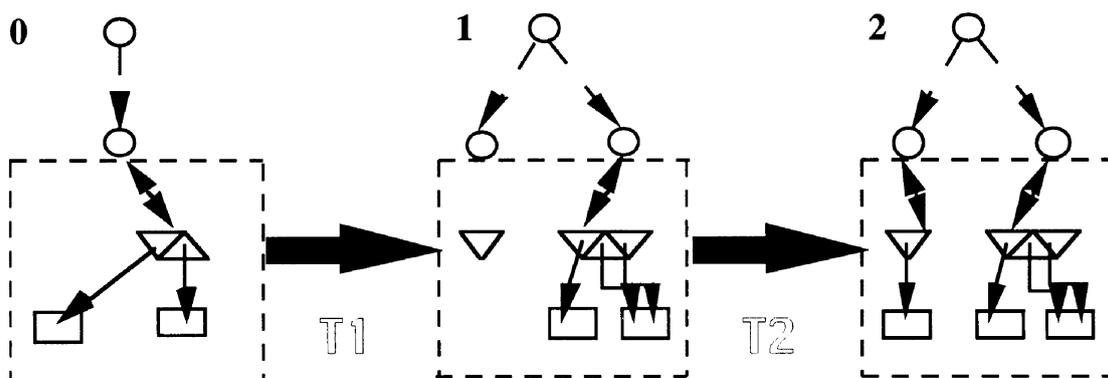


Figure 3 Transformations between document states.

Transformation processes are already clearly identifiable in the multiple media production techniques such as film production (Bloedow, 1991) and traditional cartoon animation production (White, 1986). As part of these multiple media techniques transformations take place between different physical as well as abstract media, for example the transfer of character drawings to film as well as the drawing of character sketches on the basis of a script outline. In the case of multimedia, preliminary stages may employ elements held in other physical media pending transfer to the digital.

Following such early transfers between physical media, it is assumed that the transformations that act on abstract media elements are executed by the production participants, with some assistance from tools that allow the manipulation of digitally based elements. Only a partial set of transformations will be within the technical capabilities of automatic digital tools presently available or envisaged.

Thus multimedia production can be seen as a series of transformations executed by the designers and developers participating in the process.

Revisions to the earlier production model allow the incorporation of specific media elements and their relationships with other components. The new version provides the context for media transformations, executed by the process participants, to function as the essential means for moving from artefact state to artefact state. In its final form the process model incorporates a fourth stage involving the construction of a *navigable discourse structure* as a means of defining the actual discourse structure of the document. As such this *navigable discourse structure* can be related to the intended discourse structure generated initially.

3 CATEGORIES OF TRANSFORMATIONS

Transformation activities require the manipulation of media elements. Transformation involves the generation or regeneration of one or more abstract media elements of the same or different types held on the digital medium, or possibly on some alternative medium during early cycles of design. In any state an artefact or document will incorporate a number of discourse components. The disposition of these components to one media type or another, the internal composition of the media elements, and their manner of presentation will be the result of some set of transformations performed upon elements in earlier states. The nature and number of these elements will determine the possible transformations that may be executed in order to reach the next state. The transformations fall into the two general categories of *constructive* and *supportive* transformations. The former facilitate the primary processes that directly result in elements of the final artefact, the latter facilitate subsidiary activities involving the use of elements in some kind of supporting or subsidiary role.

Manipulation of an initial text used for a multimedia document such as the *viewerdemo* offers simple examples of both categories. Transcription of the original demonstration commentary from a sound recording to a text involves a constructive transformation because at least some part of this text is likely to survive within a text element of the final presentation. At the same time this transcribed text forms the basis for a new spoken commentary which will also be part of the final document. Until this new commentary is recorded, the transcribed text will substitute for it within the production process whenever it provides essential content information relevant to the production of other media elements, such as the capture of images from the running application. Use of this transcription in such a subsidiary role represents a supportive transformation.

The supportive category also introduces a notion of purpose. It is open to production participants to decide the possible uses of a particular element, thus determining whether it is used when a particular supportive transformation takes

place. This opens the way for transformations to be used prescriptively as well as descriptively. In the last example, for instance, a participant fixed the text as the basis for the final spoken commentary by defining its generation specifically as the creation of a substitute for that commentary.

The necessity for such functionally supporting roles is one reason why transformations do not necessarily take place singly. New individual elements may be the result of more than one constructive transformation, for example the revision and merger of two graphic elements, or some combination of constructive and supportive. The notion of combining transformations is examined further below.

4 DEFINITION OF TRANSFORMATIONS

This section provides definitions of the specific transformations within each category, accompanied by general examples of their products {shown in brackets}. The analyses of animation production and the *viewerdemo* which follow provide detailed examples of their use singly or in combinations as a means of production description.

Constructive transformations

origination : the initial creation of a single medium element, without specifiable input sources or the designation of a primary source {A statement of an initial concept or a primary source document},

amplification : expansion of one element to form one or more elements in the same medium {Addition of new features to a diagram},

revision : an element (or elements) in one medium supplanting an element in the same medium via any type of revision, alteration, or subdivision {Alteration of images after comparison with what is to be an accompanying text},

translation : an element in one medium replacing an element in another {Moving images described in text},

outline : abbreviated or précis version of an element based upon another existing or as yet unrealised element in the same medium {List of functions shown in a software demonstration},

merger : combination of two elements of the same medium to form a third, also of the same type {Any composite image created from more than one source},

amalgamation : combination of at least two elements of different media types in a composite element retaining individual identities but with combined purpose {Text and images within a multimedia spread},

proxy creation : creation of an element in one medium to stand in place of an element in another medium pending the creation of that second element and representing some essential characteristics of it {Text of spoken commentary},

substitute creation : creation of an element in one medium to stand in place of an element already realised in another medium, and representing some essential characteristics of the element {Any representation of timing information}.

Supportive transformations

proxy use : employment of proxy already created {Image generation guided by text of expected commentary},

substitute use : employment of substitute already created {Use of a timing diagram to support any constructive transformation},

comparate use : use of an element as the basis for a comparison between it and another element, with a view to some constructive transformation to the latter {Fixed image sequence used to check text}.

5 ANALYSIS VIA TRANSFORMATIONS

This section shows the descriptive function of media transformations, defined in the previous section, as applied both to traditional multiple media and to multimedia production .

An analysis of the production sequence for animation shows how these transformations can be applied to traditional multiple media development. In this example, shown in Table 1 below, the stages (0 - 6) are the first seven of the sixteen detailed by White (White, 1986); the remainder are Line tests, Clean up, Trace and paint, Backgrounds, Checking, Final shoot, Rushes, Dubbing and Answer print.

In different ways the application of these transformation categories to multimedia production is both simpler and more complicated than their application to multiple media. It is less complex because, except in the preliminary stages, it is less important to take account of any changes of physical media that may be taking place. Although omitted from the preceding analysis of animation production, consideration of these carrier media would be essential for any full description of the process from the point of view of design. The different materials and techniques required, for example to prepare storyboards (Stage 2 in Table 1) and final images on acetate (Stage 9), also determine the skills which the participant directing production, fulfilling any editorial role, must co-ordinate.

On the other hand multiple media production is simpler because it need not be concerned with access operations, these being predetermined by the mechanisms used for manufacture and display of moving images on photographic film. The multimedia document will have no such standard access operations, making their choice and implementation an essential part of production, and making the application of transformation categories more complex. The analysis of the viewerdemo production which follows shows how this can be done.

The general order of production activities for the $\sqrt{\text{viewer}}\text{demo}$ followed five stages:

- 0-1** Establishment of content,
 - 2-6** Creation of two sets of key media elements,
 - 7** Construction and testing of access mechanism,
 - 8-11** Merger of key media elements and creation of additional media elements,
 - 12-14** Integration of media and mechanism and their testing.
-

Table 1 Transformations within multiple media animation production (Stages 0-6)

0	Original concept
0.1	<i>origination</i> of speech (or text)
0.2	<i>outline</i> created for final whole
0.3	<i>initial proxy</i> created for final whole.
1	Script
1.1	<i>translation</i> of speech in 0 to text
1.2	<i>amplification</i> of text in 1.1
1.3	<i>proxy</i> in 0 used, and displaced
1.4	<i>proxy</i> created in 1.2 for speech, sound and moving images in final whole
2	Storyboard
2.1	<i>translation</i> of speech in 0 into still images
2.2	<i>translation</i> of text in 1 into still images
2.3	<i>merger</i> based on 2.1 and 2.2
2.4	<i>amplification</i> (and origination) based on 2.3
2.5	<i>proxy</i> created in 2.4 for moving images in final whole
3	Soundtrack
3.1	<i>translation</i> of text in 1 into speech and sound to form part of a principal component of the final whole
4	Track breakdown
4.1	<i>translation</i> of speech and sound in 3 to graphics and text representation
4.2	<i>substitute</i> created for soundtrack in 4.1
5	Character designs
5.1	<i>translation</i> of text in 1 into still images
5.2	<i>revision</i> and <i>amplification</i> of still images in 2
5.3	<i>use of proxies</i> from 1 and 2
5.4	<i>merger</i> based on 5.1 and 5.2
6	Leica reel
6.1	<i>revision</i> and <i>amplification</i> of still images in 2
6.2	<i>substitute</i> from 4 used in place of soundtrack
6.3	<i>translation</i> of 6.1 into moving images using 6.2 as compare

Table 2 Media transformations in the production of the $\sqrt{\text{viewer}}$ demo (Stages 0-5)

0	Original documents
0.1	<i>origination</i> (in some pre-production process) of published papers describing the $\sqrt{\text{viewer}}$ $[t,g,i]$,
0.2	<i>origination</i> (in some pre-production process) of transparencies used in seminar explaining VOSE $[g,i,t]$.
1	Live demonstration
1.1	<i>origination</i> of recorded commentary $[sp]$.
2	Transcription
2.1	<i>translation</i> from recorded commentary in 1 to text transcription $[t]$,
2.2	<i>substitute created</i> for original commentary.
3	Screen dumps from repeated demonstration run
3.1	<i>origination</i> of set of screen dumps $[i]$ from a run of the $\sqrt{\text{viewer}}$,
3.2	<i>substitute use</i> of text transcription from 2 as a guide for the transformation in 3.1,
3.3	<i>revision</i> of text transcript $[t,g]$ from 2 with embedded notes indicating position of principal screen changes.
4	Screen images for multimedia demo
4.1	<i>revision</i> of screen dumps $[i]$ from 3,
4.2	<i>revision</i> of images etc. in transparencies $[i,g]$ from 1
4.3	<i>merger</i> of individual dumps from 4 to form initial set of demo images $[i,g]$,
4.4	<i>substitute creation</i> of demo image list $[t]$
5	Text keyed to images
5.1	<i>revision</i> of text from transcription in 2 to form script $[t,g]$ providing marked blocks of text,
5.2	<i>use of substitute</i> image list created in 4,
5.3	<i>amalgamation</i> of text from 5.1 with image symbols from 4.4 to show relationship between images and text $[t,g]$,
5.4	<i>creation of proxy</i> for final commentary $[t]$

In Table 2 letters in square brackets stand for the constituent media produced by each transformation $[t$ ext, sp eech, i mage, g raphics, v ideo]. The computer applications involved, most importantly the $\sqrt{\text{viewer}}$ itself and Director, might be included in this table, in terms of text or numeric files in non-natural languages, abbreviated to *plt* for programming language text; they are excluded because the purpose of the analysis is to show the transformations, their sources and products, not the means by which each came about. Table 2 details Stages 0-5; Stages 6-14

involved Demo mock-up, Access mechanism construction and testing, Formatted script, Image and script merger, Commentary recording, Script with timing, First integration, Test of first integration, Final integration.

Retrospective generation of this analysis, on the basis of a log which lacked any systematic means of recording activities, shows the robustness of the categories proposed and their usefulness in the context of multimedia production. This exercise suggested that improvements might be made in two ways, by a more succinct shorthand for recording and analysis and by a simple graphic representation. The next section considers the first possibility; the second forms an important part of on-going work.

6 RULES FOR TRANSFORMATION REPRESENTATION

Any representation for media transformations must be able to function as a means both for recording production activity that has already taken place and for planning the sequence of activity required to produce a particular document. The definitions set in Section 5 above need extension in two ways in order to provide such a representation:

- ~ identification of the sources and outputs required for each transformation,
- ~ specification of combinations of supportive and constructive transformations that produce new media elements.

The first of these extensions is essential for tracing the origin of media elements in any record of production and the second for direction of any new production. The set of transformation representation rules (TRR) shown in Table 3 uses Backus-Naur Form to define a syntax for describing media transformations in a way that meets these requirements. Reserved expressions appear in boldface type; constants appear between single quotes; names and numbers to be fixed by the user appear in italics; braces {} are used to denote groups which may be repeated zero or more times; and a vertical bar | denotes choice.

Rules 1 - 4 derive from earlier definitions of media and transformation, plus two elaborations: inclusion of program language text (*plt*) and other sign systems (*oss*) to replace the 'other' media type, and a definition of 'element compositions' which combine more than one media type. Rules 5 - 11 provide the means for identifying the sources and outputs of transformations.

Rules 12 - 14 define higher level constructs. A *complete transformation* (14) comprises either a *primary* transformation (12) or a *primary* transformation supported by a *subsidiary* transformation (13). Primary and subsidiary transformations involve specified media elements in constructive and supportive transformations respectively. The *subsidiary* transformations may be composed of

multiple supportive transformations. The positioning of the symbol '>' and the separator ',' aids differentiation between primary and subsidiary.

Table 3 Transformation Representation Rules (TRR)

<i>1</i>	<code>abstract_media_type ::= t sp s g i v plt oss</code>
<i>2</i>	<code>element_composition ::= '[' abstract_media_type { ',' abstract_media_type } ']'</code>
<i>3</i>	<code>constructive_transformation ::= orig amp rev trans outl merg amalg cprox csubs</code>
<i>4</i>	<code>supportive_transformation ::= uprox usubs comp</code>
<i>5.1</i>	<code>orig ::= 'orig >' primary source</code>
<i>5.2</i>	<code>amp ::= source_reference 'amp >' output</code>
<i>5.3</i>	<code>rev ::= source_reference 'rev >' output</code>
<i>5.4</i>	<code>trans ::= source_reference 'trans >' output</code>
<i>5.5</i>	<code>outl ::= source_reference 'outl >' output</code>
<i>5.6</i>	<code>merg ::= source_reference source_reference { source_reference } 'merg >' output</code>
<i>5.7</i>	<code>amalg ::= source_reference source_reference { source_reference } 'amalg >' output</code>
<i>5.8</i>	<code>cprox ::= proxy_name 'cprox >' source_reference</code>
<i>5.9</i>	<code>csubs ::= source_reference 'csubs >' subs</code>
<i>6.1</i>	<code>uprox ::= 'uprox' source_reference '>'</code>
<i>6.2</i>	<code>usubs ::= 'usubs' subs '>'</code>
<i>6.3</i>	<code>comp ::= source_reference 'comp' source_reference '>'</code>
<i>7</i>	<code>output ::= output_ref_number media_element_name element_composition</code>
<i>8</i>	<code>subs ::= output_ref_number substitute_name element_composition</code>
<i>9</i>	<code>primary source ::= output_ref_number name_of_primary_source element_composition</code>
<i>10</i>	<code>output_ref_number ::= stage_number output_number</code>
<i>11</i>	<code>source_reference ::= output_ref_number</code>
<i>12</i>	<code>primary ::= constructive_transformation { ',' constructive_transformation }</code>
<i>13</i>	<code>subsidiary ::= supportive_transformation { ',' supportive_transformation }</code>
<i>14</i>	<code>complete_transformation ::= primary subsidiary ',' primary</code>

Use of TRR provides a more succinct description of the production process, defining more clearly the relationships between activities, their sources and

products. Production of the $\sqrt{\text{viewer}}\text{demo}$ can now be recorded fully in the form shown in Table 4. Output reference numbers are highlighted using boldface type to aid backwards traceability. This visual aid is useful because the syntax does not guarantee any specific mapping between sources and outputs of different transformations.

Table 4 Production of the $\sqrt{\text{viewer}}\text{demo}$ following TRR

orig > **01** published papers [*t,g,i*]
orig > **02** seminar transparencies [*g,i,t*]
orig > **11** recorded commentary [*sp*]
 11 *trans* > **21** transcription [*t*]
 11 *csubs* > **21** transcription [*t*]
usubs 21 >, *orig* > **31** screen dumps [*i*]
 31 *comp* 21 >, 21 *rev* > **32** transcript + screen change notes [*t,g*]
 31 *rev* > **41** screen dumps [*i*]
 02 *rev* > **42** transparency images [*i,g*]
 41 42 *merg* > **43** initial set of demo images [*i,g*]
 43 *csubs* > **44** demo image list [*t*]
usubs 44 >, 21 *rev* > **51** script in marked blocks of text [*t,g*],
 51 44 *amalg* > **52** script + image symbols [*t,g*]
 final commentary *cprox* > **52** [*t*]
uprox 52 >, 52 *comp* 43 >, 43 *rev* > **61** demo mock-up images [*i,v*]
 61 *outl* > **71** dummy elements [*i,v,t,sp*]
 final elements *cprox* > **71**
orig > **72** access operations [*plt*]
uprox 71 >, 71 72 *amalg* > **73** test demo [*i,v,t,sp,plt*]
 complete demo document *cprox* > **73**
uprox 73 >, 72 *rev* > **74** tested access operations [*plt*]
 52 *rev* > **81** script in final presentational form [*t*]
 61 81 *amalg* > **91** first text/image combination [*t,i,v*]
 91 *rev* > **92** text inconsistency check [*t,i,v*]
 92 *rev* > **93** image inconsistency check [*t,i,v*]
 93 *trans* > **101** recorded commentary [*sp*]
 101 *outl* > **111** timing information [*t,g*]
 111 93 *amalg* > **112** script + timing information [*t,g*]
 final commentary *cprox* > 112
 74 93 101 *amalg* > **121** first integration [*i,v,t,sp,plt*]
uprox 112 >, 112 *comp* 121 >, 121 *rev* > **131** final
 revision [*i,v,t,sp,plt*]

7 CONCLUSIONS

The novel concept of media transformations provides the means of showing how the document structure may move between static states as its scope and form are extended. The differentiation made between constructive and supportive transformations recognises that, among the interim products of production, some contribute directly to the final document, while others act in a subsidiary capacity. The media transformation that are defined provide a comprehensive guide to production activities and a detailed means of describing, communicating and proscribing the production of any particular document. Analyses of animation production and the document produced in the *viewerdemo* case study, show the applicability of the concept to both multiple media and multimedia. A set of transformation representation rules (TRR) in BNF enables the succinct description of production. Composite statements will show the complex combinations of source elements and transformations that may result in a single new media element. The major concern of future work will be the visualisation of production in terms of transformations, providing a graphical representation to ease communication between participants for all purposes.

8 ACKNOWLEDGEMENTS

The author acknowledges gratefully the overall guidance and assistance of Anthony Finkelstein and the financial assistance of the Engineering and Physical Sciences Research Council of the UK during the time of this research.

9 REFERENCES

- Bloedow, J. (1991) *Filmmaking foundations*. Focal Press, Boston.
- Carroll, J.M. and Moran, T.P. (1991) Introduction to special issue on design rationale. *Human-Computer Interaction*, **6**, 197-200.
- Kramer, J., Magee, J., Ng, K. and Sloman, S. (1993) The System Architect's Assistant for design and construction of distributed systems, in *Proceedings of Fourth IEEE Workshop on Future Trends of Distributed Computing Systems*, Lisbon, September 1993. IEEE Computer Society Press, Los Alamitos, Ca.
- Macromedia Inc. (1994) *Macromedia Director Version 4 - Using Director*. Macromedia, San Francisco.
- Morris, S.J. and Finkelstein, A.C.W. (1993) Development of Multiple Media Documents, in *COOCS 93, Proceedings of ACM Conference on Organizational Computing Systems*, Milpitas Ca., November 1993. ACM Press, New York.
- Morris, S.J. and Finkelstein, A.C.W. (1996) Integrating design and development in the production of multimedia documents, in *Proceedings of MMSD 96*,

International Workshop on Multimedia Software Development, Berlin, March 1996. IEEE Press, Los Alamitos Ca.

Nuseibeh, B.A. and Finkelstein, A.C.W. (1992) Viewpoints: A vehicle for method and tool integration, in *Proceedings of CASE '92*, Montreal, Canada, July 1992.

White, T. (1986) *The animators workbook*. Watson-Guptill, New York.

10 BIOGRAPHY

Stephen Morris originally studied the history of art at Cambridge University and subsequently followed a career in town planning. He later returned to higher education and obtained his doctorate at the Department of Computing of Imperial College. He now holds the post of Lecturer at City University in London. His research interests include the investigation of the theoretical and practical problems associated with the cooperative production of multimedia documents, in particular the implications for software engineering of integrating the separate activities involved in content generation and software development. He also works on representational languages for object-oriented software development.