The programming-like-analysis of an innovative media tool

Chris Roast, Elizabeth Uruchurtu, Andy Dearden,

Communication and Computing Research Centre
Sheffield Hallam University
153 Arundel Street, Sheffield, S1 2NU, UK,
+44(0)1142255555
c.r.roast@shu.ac.uk

Keywords: Automating media production, Post-production, automation, disruptive technology, user centered evaluation.

Abstract

This paper describes a case study in which evaluation techniques have been developed and applied to a novel commercially developed tool for supporting efficiency and effectiveness of a digital film production processes. The tool is based upon a familiar concept in digital publishing that of separating style from content, and as such, it represents a challenge for intended end users since it moves them away from traditional working practices and towards programming-like-activity. Two alternative user interfaces have been developed following a commercial development route. Approaches to analyzing the effectiveness of the tool and its interfaces prior to its widespread adoption are described and the conclusions from this analysis are illustrated and discussed.

1. Introduction

This paper is concerned with the use of the analysis and evaluation techniques in a commercial design and development setting. It describes a case study of applying techniques in the dynamic context of commercial software development, and focuses upon a support tool developed for digital media production market. The work reflects some of the complexities of working closely with commercial development teams where external factors impact upon development resources and priorities, subsequently this impacts upon the incorporation of new evaluation techniques.

The tool is one of a set of related software tools that have been developed with the specific aim of simplifying and re-configuring the activities of media production and publication within the digital film industry, to reduce costs and improve quality. One example of this is the potential to reduce the duplication of work, by supporting commonly repeated activities. For example, when similar graphic design and layout work is required for a variety of audience languages in, say, a DVD menu, the need for repeated re-design for each language could be eliminated. On the face of it, this is a relatively straight forward concept. However, the reality of tool design and adoption is complicated by having to be integrated with existing work practices. In particular the tool under consideration in this paper is intrinsically "disruptive" in that it presents an innovative opportunity to deliver a step change in efficiency. The innovation is unfamiliar to the users and involves practices that are qualitatively different to existing work activities. Standard models of the diffusion of innovations suggest that the decision to adopt an innovation is dependent on perceived ease of use and perceived benefit (Rogers 2003). Hence, although these new tools represent an opportunity to optimise work, adoption of the technology is far from assured.

In terms of the practicalities of user adoption it was noted that the user would need to focus their attention upon manipulating information structures that were in principle abstract. However, current use is strongly focused upon concrete material outputs. This motivated the aim of examining the tool in terms concerning human factors in programming.

2. The Tool Concept

The working context of this research is that of digital film and related media production, where international distribution requirements demand regional contributions, such as adaption of publicity material to suit different languages, or providing subtitles and/or audio dubbing to a title. Complementing the demand for regional contributions, there is the commissioning studio's desire to centrally manage and control overall quality and maintain brand identity.

As an example, consider post-production for film distribution in formats such as blu-ray and DVD. These often involve sophisticated design work in areas such as the interactive menus and general viewer interaction. Such design must adhere to studio standards regarding details such as menu complexity and interactive structure. The first instance of such design is completed in a native language and style of the publishing studio, reflecting the brand of the title.Once that design is approved, the same design work needs to be done for the full range of languages to be supported. The process of building and quality assuring the full design is potentially very complex. It involves the textual, visual and video content associated with different languages and regions to be brought together and combined to work as a coherent whole. Advanced tool support allows native single region/language designs to be imported and used to build and configure templates that are then capable of defining how arbitrary textual and visual assets can be used to generate a version covering all the required languages.

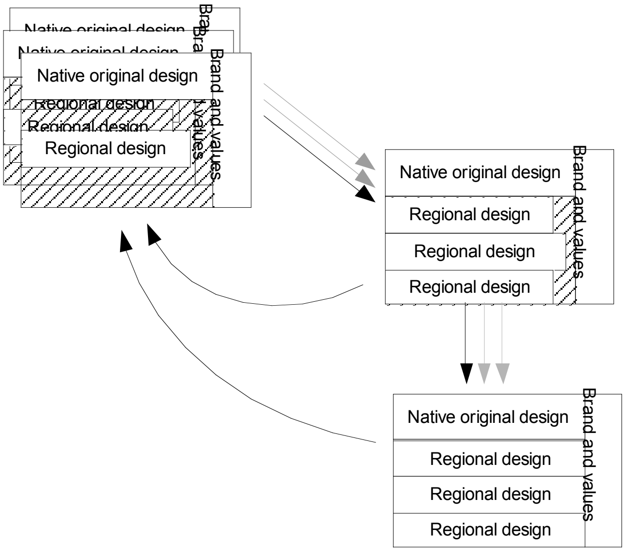

Figure 1. The existing processes.

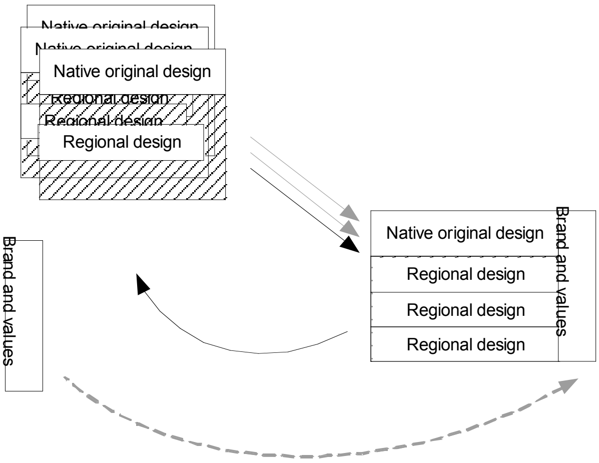

Figure 2. The proposed tool enhanced process.

The significant transformation enabled by the tool is to reduce the effort of breaking-up the work into independent components and then re-combining them. This strengthens the potential for centrally managing product quality, while also reducing overall effort through automation. Figures 1 and 2 illustrate this. Figure 1 shows the exiting process: the initial native version of a film title (or its related collateral), is given out to regional offices. The regional variants attempt to meet the given brand and values, however this requires central confirmation. When returned, each element is checked and revisions may be required at that point. Once all variants are integrated into a final product further checking is necessary and more revisions may be required for specific elements. Clearly, inefficiencies arise when the integrated result has to be checked for overall quality, and when regional work has to be re-done to address any issues raised. Figure 2 shows the potential tool supported improvement, the branding and quality values of the initial native version are captured via abstractions such as templates and rules so that regional work is less able to disrupt quality. Once regional inputs are integrated the branding and values can be applied uniformly to all contributions, thus reducing the burden of quality checking and reducing the likelihood of having to re-work content.

2.1 The general architecture

The general architecture adopted by our industrial partner is to provide a consistent structure for the artefacts that can be imported, manipulated, processed and integrated. This then allows for generalities to be aligned to the structure. The structure used is hierarchical, with the lowest level objects in the hierarchy representing individual assets such as images or textual elements, and higher level objects grouping those beneath them.

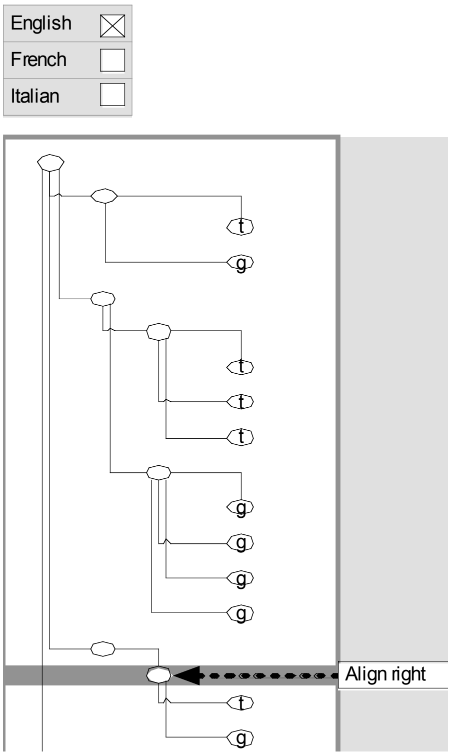

Figure 3. The node based user interface illustrated.

Generality is achieved by allowing variability in what a structured object can produce. This is expressed in terms of rules associated with nodes in the hierarchy. The rules fall into a number of categories including:

- Rules that apply purely to textual features. These include rules that set or scale fonts sizes, change font faces, etc. E.g. "Set the font size on the associated cell to 12pt."
- Rules that apply purely to image-based assets. These include scaling images, replacing a default image with an alternative. E.g. "Set the height of this image to 10mm maintaining aspect ratio."
- Rules that relate, move or align nodes both relatively and absolutely. E.g.: "Set the width of this node equal to node NODENAME."; "Align offspring to the right.", and "Move left 8px."

In addition to associating a rule with a node, the rule can be qualified with respect to regional language. Hence, the same node could have rules that are used only for specific language versions. For instance a single node might have its font set to 12pt, when using the French or Flemish translation; 10pt in Spanish; and uppercase in, say, Turkish.

The node hierarchy and the rules associated with nodes are used to generate all the required versions of the product specific to each language. Specific language translations are automatically incorporated via another service. Hence, from one structured object, numerous region-specific versions of the original native one can be specified and generated. For a product such as a blu-ray menu the language versions generated are integrated to form a complete multi-lingual menu.

2.2 The user interface alternatives

Within this research two interfaces ("Node Based" and "Process Based") were available. These provided different means by which the users are able to view and manipulate the structured document and its rules.

The Node Based User Interface In the node based interface the node hierarchy is the primary means of viewing a configuration. Conventional hierarchical structure management is supported, allowing parent nodes to be "folded" and "unfolded", and node type information (textual or graphical) is evident from the leaf node icons.

In order to see what rules are used on a particular node the user has to "select" that node, then the rules associated with it are shown. Rules can also be added or deleted on a node by node basis. The language specificity of a rule is indicated by checkboxes at the top of the display. Figure 3 provides an illustration of this, in which a node is shown to have an "align right" rule that is applicable for the English language.

The Process Based User Interface In the process based interface the node hierarchy is visible in the same way. In addition, a region of the display is used to show all the rules being used (for all the nodes). The rules are shown in the sequence in which they are used. Figure 4 provides an illustration of this. Each rule can be unfolded to see details which include the nodes it is applied to and for what languages. The figure shows also that when one of the nodes is selected its position in the hierarchy is also highlighted. Specifically, the illustration shows a scale font rule applied to three nodes for two languages cases (UK English and USA English).

When we focused upon the analysis of the tool it was recognised that its adoption, or its ease of adoption, is highly dependent upon the interface used. In order to engage stakeholders in the assessment of the tool, we were able to focus upon the tool features differentiated by the two interfaces to promote greater critical insight.

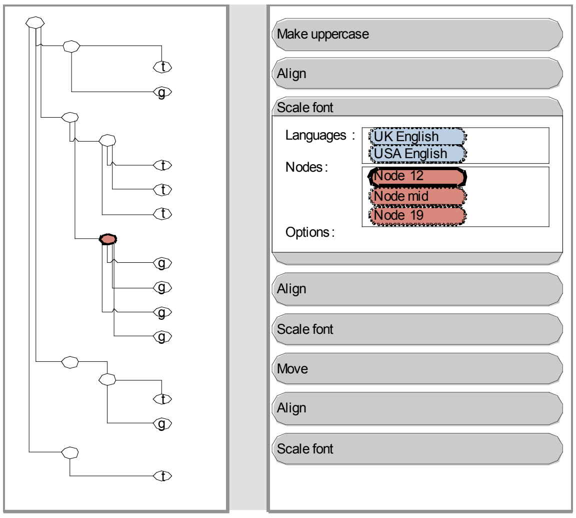

Figure 4. The process based user interface illustrated

3. Analysing Context and Users

As mentioned earlier, the value of adopting this tool concept is reliant upon tool users employing it as intended. The tool represents a significant new activity for intended end users and has the potential to disrupt established patterns of work and runs the risk of not fulfilling its aims. In particular the two alternative user interfaces provide routes to successful adoption that are different. In order to explore and understand such issues, the research team conducted interviews, observation and workshops with: a small group of early adopters; potential users and their managers; and also requirements analysts working for the tool developers. The aim of this process was to establish an understanding of the factors and values influencing and supporting changes of practice within end user work contexts.

This analysis involved initially characterising the intended users of the tool. Lead by this observation and our engagement with intended end users, we employed lightweight domain mapping to identify potential issues and profile potential end users. These provided a means of focusing our analysis and facilitating our consultation with stakeholders. Following this a comparative assessment of the tool interfaces was conducted in consultation with stakeholders. Two approaches were used in this assessment based upon the fact that the tool functionality in effect required intended users to access and use programming constructs. Both program comprehension assessment and the Cognitive Dimensions framework were employed (Blackwell, 2006, Clarke, 2001, Dearden, et al 2003, Green and Blackwell 1998).

3.1 The users in context

The intended users are primarily professional graphic designers working in a digital context. They are commonly accredited or highly experienced in using professional graphics, layout and typesetting tools such as Photoshop and InDesign. Their experience and expertise in using such tools to produce high quality static graphics is highly valued.

Within the broad setting of digital media production individual roles, responsibilities and work flows are well established. In particular there is evidence that professionals are very "process aware" and are well aware of the implication of delays, errors and poor quality work on the overall production process. Hence although work is clearly demarcated, overall product quality and efficiency of production appear to be collectively understood. Thus, the perceived value of the tool should be positive.

In addition to their professional expertise and attitudes, it is also of value to characterise the manner in which they work. In terms of psychological descriptions, they work in a highly concrete craft-like manner - what could be termed "hands-on" and working "by-eye". The values they prioritise in their work are those of visual precision in colour and layout. Their primary work tools emphasise direct manipulation: immediate visual feedback, responsiveness and visual representation of the final product. One of the terms often used in referring to design quality is to ensure that the result of their work is "pixel perfect". As an illustrative example, it was observed that users were happier to align objects by eye than by employing build-in object alignment options.

From this perspective it can be seen that the nature of the tool examined in this paper is likely to be challenging for these users. The node hierarchy and its rules are the object worked-on and it is an object capable of generating many products - one for each language used. In this setting the notion of "pixel perfect" is not immediately relevant.

The issues faced by individuals in adopting an innovation are characterised by Rogers (2003) in terms of: relative advantage; compatibility with existing practice; complexity of use; ease of trialling; the ease of observing and demonstrating the value. In the following analysis of the tool we are able to focus on the first two of these. The remaining three were governed by other factors at the organisational level and were not the primary focus of the academic collaboration.

3.2 Domain mapping

Our approach to domain mapping is based on Ontological Sketch Modelling (see Blandford et al, 2004). With this approach the concepts and attributes of a domain and a tool are identified and classified in terms of their relevance to intended users as well as the quality with which they are supported by a system and its interface. In the same analysis, actions upon concepts and attributes can be classified in terms of their ease and inter-relationships.

For intended tool users their conceptual model is taken to be close to that evident from their expertise with existing graphic design tools. Hence we conclude that users are competent in understanding and working with the following objects and actions:

The Canvas; Layers that can be promoted, demoted, created, deleted, merged, grouped and ungroup; Styles that can be created, modified, deleted, applied; Tags that can assigned or un-assigned; Regions that can be selected, cleared, tagged or untagged; and so forth.

The key insight from this is that the concepts users are already familiar with map closely to some of those assumed in the tool. For example, the layer hierarchy supported in drawing tools is analogous to the node hierarchy provided in the new tool. The impact of introducing the new tool to users with this knowledge can thus be assessed in terms of what additional conceptual understanding the tool demands of them. Two key concepts apparent are:

- The introduction of regional languages. Prior to the new tool, work on the same project from different language settings would be treated as different "jobs" (with some common elements). The tool enables languages to be encoded into the work, hence the intended users view is one of having a project file that is capable of supporting several specific jobs.
- The introduction of explicit rules. Prior to the new tool the facilities closest to those offered by the tool would be styles, macros and plug-ins. With the new tool the user is able to, and expected to, assign and configure sets of rules as a means of achieving a consistent style for a number of jobs within a project.

Hence, in terms of the mapping analysis the tool introduces a new level of abstraction that we'll term Project. A project embodies a set of jobs that have common purpose but vary with respect to language and region. The concept of language and region now has a formal representation with the system. In addition a project includes the new concept of a RuleSet, and the concepts used to express rules. The particular concepts used to express rules were all determined from specific user needs and therefore were not critically assessed at an individual level. Having said that, it was observed that the majority of rules were procedural in nature, focusing upon what needs to be done as opposed to the result of doing it.

3.3 Segmenting user types

At a technical level the introduction of rules and support for cross-language design are the key new concepts. As a result the users are required to become familiar with the programming-like concepts of rules and the generality (across languages) that they are able to offer. In addition to being familiar with hand crafting a set of concrete images in the native language for a specific purpose (such as forming a blu-ray menu set), the user is required to configure a set of rules so that the same design quality is achieved for language specific alternatives for the same project. In order to analyse how the balance between native-language crafted design and more general cross-language rule configuration, we compare two analytic frameworks for characterising usability claims embodied in the designs. These claims can then serve as a basis for engaging with tool users.

To frame that analysis we consider three archetypal user mindsets that represent possible ways users are likely to view the tool.

- **Keep it simple**. This user mindset characterises users who make do with the tool. Although working with the rules offered, they are not confident in working in the abstract terms provided by rules. So for example, although a rule may exist to align a group of nodes, this type of user would be more than happy to perform the alignment manually. If asked, they would say they were working on getting a specific image "right" (see Khazaei and Roast, 2003).
- **Tool proficient**. This user mindset characterises users who view the tool as one of the many they need to use as part of their practice and therefore work at being effective with it. Given a particular project and wishing to achieve a particular effect, they identify and apply an appropriate rule. Hence they use rules but are not working towards elegant configurations of rules to capture good design. If asked, they would say they were working on (i) a known set of images that are formed by rules and (ii) getting the rules correct for the images required.
- **Keep it general**. This user mindset is mirror of that intended by the tool developers. The tool adopter understands that: (i) some initial work is traditional and focused on developing quality graphics for the native language component of a project; (ii) subsequent work is aimed at capturing that design quality in more generic terms. We could characterise them as individuals who might "re-factor" a rule configuration to minimise unnecessary repetition and localise information. If asked, they would say they were working on (i) a set of images that are formed by rules and (ii) getting the rules correct for the known images and those that may yet be required by the project.

Interviews with users and user representatives validated these characterisations and indicated that the predominant user population were "Keep it simple" - characterised as not "getting it". The preferred user type was the "Tool Proficient" and these were seen as the most likely feasible target user. The "Keep it general" user type was recognised as possibility but unlikely because of the risk of time being misspent on preparing and not "doing".

An additional factor limiting progression beyond "Tool Proficient" was the perceived risk of working on rules only to find that they are not operating as required and other rule sets would be more appropriate. The user priority is focused upon the quality of the outputs generated and not on the means by which it is achieved. In short, the traditional way of working is known, and known to work. Hence the pay-off working with the tool needs to be easily realised.

Functional information focuses upon information about the overall goal of a program. Thus it is not specific to technologies or notations. For post-production this information concerns all the outputs from a specific configuration being at the expected standard. Q: "Is the film title always displayed on the menu and does it span the screen in all languages?"
State information focuses upon the state of variables and objects at particular points in program execution. For post-production this concerns information regarding the positioning, scaling, etc, of the graphical and textual elements that go to form an image or a set of images. Q: "After positioning image1, what's the position of the image2?"
Control Flow information focuses upon information relating to sequence of activities and events that occur in a program. For post-production this concerns when more than one rule is used and the sequence by which they are used. Q: "Will node3 be scaled then aligned or aligned then scaled?"
Data Flow information focuses upon how data is passed and manipulated. For post-production this information concerns how elements and their scale and position influence other elements, by virtue of rules such as "align-left" or "scale-to-element" Q: "When node4 is moved, what other nodes are affected?"
Operation information focuses upon the specific operations that take place. For post-production this means understanding what specific rule types do. Q: "Will the Fit-to-bounding-box rule shrink and/or enlarge the font point size used in a text node?"

Table 1. Programming comprehension information types and their mapping to post-production and example questions.

4. Analysis - Human Factors in Programming

The analysis stage comparatively examined the two alternative user interfaces with a view to gaining insights about both the interfaces and the tool. Both program comprehension (Pennington, 1987, Roast and Bettle, 2001) and the Cognitive Dimensions framework (Clarke, 2001, Dearden, et al, 2003, Green and Blackwell, 1998) were used to structure the analysis. As mentioned earlier, conducting this work in a commercial setting influenced how analysis techniques could be applied. One particular impact was that the limited time with user representatives and stakeholders meant the process of engagement had to be simplified.

4.1 Developing Motivating Questions

For the program comprehension perspective Pennington's information types relevant to comprehension were used to structure engagements (Pennington 1987). For each type comprehension questions were formulated in terms relevant to the tool. Specific foci for the questions were drawn from the domain analysis, ensuring that new potentially challenging concepts were focused upon. Table 1 shows the information types considered and example domain/tool specific questions. These questions were subsequently used to drive interviews and discussions with user representatives.

A similar approach was adopted with the Cognitive Dimensions framework. The framework provides a set of valuable concepts for the assessment of complex interactive systems. In particular they have been derived from extensive research experience examining and analysing systems that involve a combination of interactive behaviour and the use of notational representations. Key tool concepts that arose from the domain mapping were used to express tool related questions based upon those recommended in Blackwell (2006) and used successfully in Roast and Khazaei (2007). These questions too served as a basis for facilitating user representative engagement in reflecting upon the value of the tool.

Focusing upon a few key concepts enabled inter-concept relations to be explored more thoroughly. The materials used for discussion and reflection supported this by following a tabular form in which differing possible relations were open for consideration. An example of this form for questions about the concept of "visibility" is shown in figure 5. This approach was adopted so as to simply present the range of alternatives that could be considered, and to avoid "leading" the assessment in a specific direction. (A criticism that could be directed at the comprehension questions developed.)

What are the dominant / common ways in which these concepts are shown together or reached from one another?

to **from**	a rule (or rule sequence)	a language (or set of languages)	a node (a group of nodes)
a rule (or rule sequence)	***collapsing rules and scrolling***	***visible***	***visible***
a language (or set of languages)	***1 click operation (filter on language)***	***check / uncheck active languages - 1 click each***	***filter on language -> rule sequence -> find all nodes***
a node (a group of nodes)	***1 click operation (filter)***	***filter on node -> rule sequence -> find each languages***	***scroll and collapse subtrees***

Figure 5. An example of the question format for examining inter-concept relations for visbility, with participant and analyst's notes shown.

Initially the user activity was assessed in terms of the Cognitive Dimensions framework by exploring the general types of activity expected. These include: "Searching" - finding information and knowhow and referencing; "Transcribing" - copying substantial amounts of information from some other source into the system; "Incremental" – repeatedly adjusting small bits; "Reorganising" - re-working solutions previously created; and, "Playing" – using the tool to explore new ideas and what's possible. In the subsequent questioning these were found to map well to the different user types discussed above.

4.2 Questioning

The analysis was conducted using the questions as drivers for discussions. These discussions involved: establishing a common understanding of question elements; and relating them where possible to the "tool proficient" user type. The exploration in some cases lead to tool demonstrations and/or walkthroughs illustrating specific behaviours, and in others it resulted in discussions about the relevance of the specific questions and related alternative questions.

An example of these would be to take a question looking at understanding how a given node will be processed in a project. The walk-through of this question using an existing project revealed what a systematic rational approach to answering it may entail. In this case: (a) identifying the node and any rules associated with it; (b) identifying nodes with rules that potentially influence it; (c) interpret the composite effect of those rules in order; (d) modify that interpretation with represent to specific languages.

Collectively this type of analysis was beneficial in revealing the complex nature of the system, and also showing the type of designs that are likely to be understood and those likely to be too complex for user. Having grasped the potential complexity of the task, the user representative is able to respond with reflections on other factors relevant to the assessment. As a consequence insights regarding: (i) tool weaknesses; (ii) question characterisation/realism; and (iii) domain specific assumptions, were forthcoming.

For this example, the user representatives were able to recognise a tool weakness that rule instances could convey more constructive and specific information about the function they perform (ie. relative or absolute transformations, absolute transformations making the interpretation process far simpler). The same question was challenged on the grounds of realism, this task was judged to be relatively uncommon and probably only applied to specific nodes. Finally, the question revealed what was emerging common practice with the tool, in that some rules were normally only applied at leaf nodes.

The main value of this analysis approach is that it provided formative feedback for both design improvements but also insights regarding the latent factors influencing effective tool use. For instance some of the feedback around realism and domain specific assumptions revealed subtle details not documented or expressed elsewhere in our assessment.

4.3 Outcomes

In general the analysis revealed a number of areas of where information support for users was most relevant. However it was worth noting that generic lead question on each of the Cognitive Dimension question tables (as shown in Figure 5) did appear to motivate additional feedback (not based purely on the concepts used for the tables). The most relevant feedback resulting from this was re-expressed for the development team in terms of guidelines based upon contrasting the two user interfaces. These included:

- Meaningful navigation and views support the user in seeing how nodes and rules are inter-related. For example, being able to see which nodes might influence another, and being able to see the set of rules that operate on those nodes. For both the node based and the process based user interface the most meaningful view is that of the hierarchy of nodes. However in the node based there is no other support for identifying related nodes or easily finding them. By contrast in the process based user interface there is functionality that allows filtering on a node name, and the automatic identification of related nodes. Thus, in this area the process based interface more effectively supports the user.
- The tool would benefit from support for interpreting the composite effect of a series of rules. This limits the need for the user to keep a maintaining the "cumulative effect" in their head when examining a specific design. For both the node based user interface and the process based user interface there is limited support for composite effects, other than familiarity with the rule names and their use. (i.e operational information).
- Rules are a central concept to the tool, most information flows and activities centre on them. Despite this, access to rule instances is complicated by: poor support for differentiating instances; and poor rule abstractions. Simple technique that could be used to help address this is to allow rule instance annotations, or comments. Despite their core importance rules cannot be introduced without being fully defined, this can be over committing for users unfamiliar with specific rules. Providing rules with sensible defaults, rapid feedback and/or expandable detail could help address this.
- The node hierarchy on which a specific project is based is largely static for that project. However, that does not mean that they are easily recognised or remembered when working on a project. Even a simple facility such as allowing the (re-)naming of a node would alleviate considerable mental effort.
- The rules are in effect highly abstract concepts, however their abstract nature (and thus power) is not promoted. If rule instances were not predicated on specific node hierarchy, their abstract nature would be clearer, as would their potential to embody knowledge about how to process some classed of assets.

4.4 Comparing Analyses

Although the two approaches to question lead analysis are hard to formally compare, it is evident that the more open leading question on the Cognitive Dimension questions (as in Figure 5), was more thought provoking for participants. In addition, despite participants having a limited time to engage

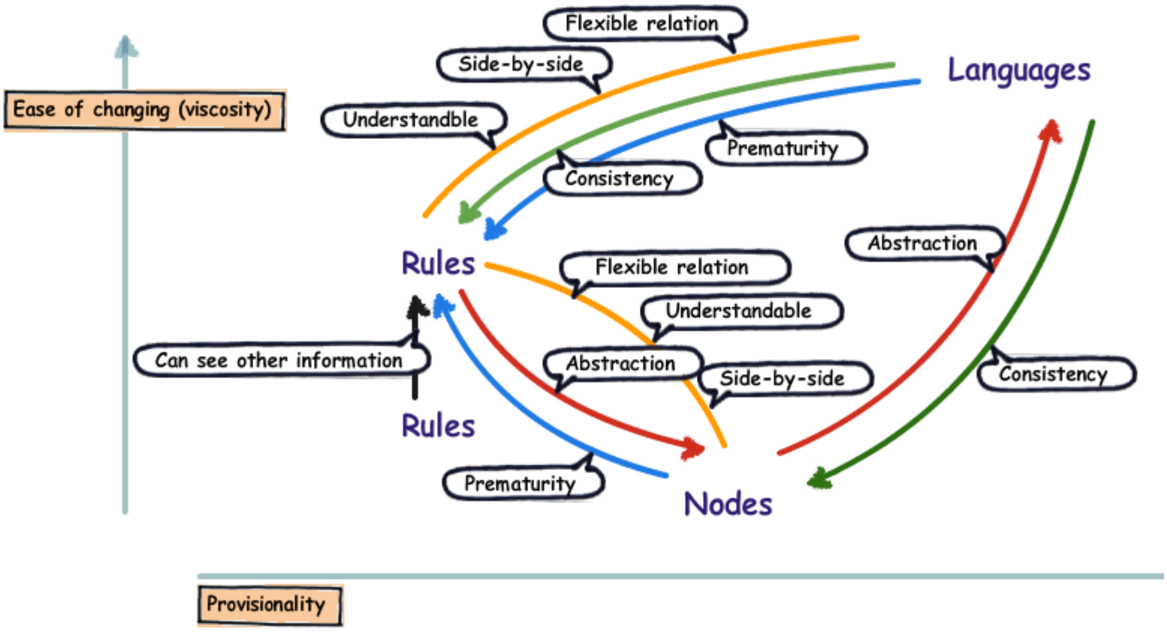

Figure 6. The lightweight summary of Cognitive Dimensional relationships for key domain concepts

with the assessment the systematically comprehensive tables based on the Cognitive Dimensions appeared to provide a structure that was open and thought provoking. The open style seems to have supported more reflection about tool support as a whole.

Using the Cognitive Dimension questions from a number of participants, the question responses were examined. This assessment was aimed at mapping the questions back to the basic concepts in an attempt to see a coherent simple picture of the assessment. The three driving concepts: nodes, languages and rules were positioned relatively on an x-axis (for provisionality) and a y-axis (for viscosity). Arcs were drawn for strong inter-concept relationships arising form the assessment. The resulting diagram is shown in figure 6. Although based on relatively limited data gathered and the simple instrument of the question grid. It is interesting to see what commonalities appear:

- The concepts of easy mental operations ("understandable"), juxapositioning ("side-by-side") and low viscosity ("flexibility") between concepts applied between Languages and Rules and Rules and Nodes.
- Consistency and prematurity were less conformant across the three concepts.
- Abstraction between the three concepts aligned with provisionality.
- The discussion of the concept of rules resulted to the reflections on rule instances being distinguished from classes of rules.

We believe this simple layout of key domain concepts and indications their cognitive dimensional inter-relatedness is a useful way of attempting to gain an overall picture of a analysis. It is hard to draw strong conclusions from such layout.

5. Conclusions

The development of advanced tools in a number of settings presents potential usability issues especially in cases where the advance demands a conceptual shift by the user. Here we've reported on the analysis of one such tool that has been developed to improve the efficiency and effectiveness in film post-production processes. In this case we've recognised the conceptual shift as being similar to empowering the end user with programming-like functionality.

The analysis of the tool has used lightweight domain modeling, user characterizations to focus a question led qualitative exploration of the tool with user representatives. This approach seems to have been particularly valuable in both engaging user representatives in critically assessing the tool and enabling formative guides about addressing key issues with the tool.

For tool support in the context where tool sophistication exceeds the simple "instrumentation" of existing user activity, new analysis approaches need to be considered. Selecting and using such methods is not simple, in our case study here, two rational approaches followed were that of employing a program comprehension framework and the Cognitive Dimensions framework. We've shown that each approach could be adapted to explore the tool under examination. In addition the approaches largely complemented each other.

6. Acknowledgments

The authors would like to acknowledge the support of the EPSRC and the Technology Strategy Board that enabled this work, and our industrial partners for participating in this research.

7. References

Blackwell, A. F., (2006) Ten years of cognitive dimensions in visual languages and computing: Guest Editor's introduction to special issue, Journal of Visual Languages & Computing, Volume 17, Issue 4, Ten Years of Cognitive Dimensions, ISSN 1045-926X, DOI: 10.1016/j.jvlc.2006.04.001.

Blandford, A.E., Green, T.R.G. and Connell, I.W. (2004). Formalising an Understanding of User-system Misfits. Proceedings of EHCI-DSVIS 2004, the 9th IFIP Working Conference on Engineering for Human-Computer Interaction (Jointly with the 11th International Workshop on Design, Specification and Verification of Interactive Systems), Hamburg, 11-13 July 2004.

Clarke, S. (2001). Evaluating a new programming language. In G. Kadoda (Ed.) Proceedings of the Thirteenth Annual Meeting of the Psychology of Programming Interest Group, 275-289.

Dearden, A. M., Siddiqi, J. & Naghsh, A., Using Cognitive Dimensions to Compare Prototyping Notations. In Proceedings of the Fifteenth Annual Meeting of the Psychology of Programming Interest Group. Keele University, UK, 8th –10th April 2003.

Green, T.R.G. and Blackwell, A.F. (1998). Design for usability using Cognitive Dimensions. Tutorial session at British Computer Society conference on Human Computer Interaction HCI'98.

Khazaei, B., and Roast, C. (2003) The influence of formal representation on solution specification in Requirements Engineering 8(1): 69-77

Pennington, N. (1987) Stimulus structures and mental representations in expert comprehension of computer programs, Cognitive Psychology.

Roast, C. and Bettle, T. (2001) The Impact of Colour on Web Application Programmers. IHM-HCI'2001 - The 15th Annual Conference of the British HCI Group (volume 2)

Roast, C. and Khazaei, B. (2007) An Investigation into the Validation of Formalised Cognitive Dimensions. Interactive Systems. Design, Specification, and Verification, Lecture Notes in Computer Science, Volume 4323/2007, 109-122, DOI: 10.1007/978-3-540-69554-7_9.

Rogers, E. M. (2003). Diffusion of innovations (5th ed.) London: Simon and Schuster.