Text-to-Movie authoring of anatomy lessons

Vaishnavi Ameya Murukutla

Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Vaishnavi Ameya MURUKUTLA

Thèse dirigée par **Olivier PALOMBI**, Professeur
et codirigée par **Rémi RONFARD**, Directeur de Recherche

préparée au sein du **Laboratoire Jean Kuntzmann** et de l'**Inria**
dans l'**École Doctorale de Mathématiques, Sciences et Technologies de l'Information, Informatique**

Text-to-Movie Authoring of Anatomy Lessons

Text-to-Movie
Création de leçons d'anatomie

Thèse soutenue publiquement le **7 décembre 2021**,
devant le jury composé de :

M. Marc Braun
Professeur, Université de Lorraine, Rapporteur

M. Pierre-Antoine Champin
Associate Professor, Université Claude Bernard Lyon 1, Rapporteur

M. Philippe Joly
Professeur, Université Toulouse III - Paul Sabatier, Examinateur

Mme. Marie-Christine Rousset
Professeur, Université Grenoble Alpes, Présidente

M. Nady Hoyek
Associate Professor,Université Claude Bernard Lyon 1, Examinateur

M. Olivier Palombi
Professeur, Université Grenoble Alpes, Directeur de thèse

M. Rémi Ronfard
Directeur de Recherche, Université Grenoble Alpes, Co-Directeur de thèse

Contents

List of Figures

List of Tables

Abstract

Anatomy is one of the most essential yet challenging subjects in medical education. It is introduced very early in the curriculum, and the student is required to retain this knowledge throughout their practice. One of the significant challenges faced in anatomy pedagogy is presenting complex three-dimensional body parts in classes. Multimedia methods such as animated videos and 3D anatomical models are popular to visualise the human body. But the issue of developing an easy to use software that enables teachers to create their own lessons without the need for design experts is ongoing. In this thesis, we present a Text-to-movie authoring system for anatomy professors. We provide the solution in three parts. We first present a formal language to describe all the visual elements in a video. This language is both human and machine-readable, and it allows us to annotate the video in text form. This Prose Storyboard Language (PSL) is developed for broader use in film direction and analysis. In the second part of the thesis, we present the anatomical specialisation of this language called the Anatomy Storyboard Language (ASL). This domain-specific language forms the basis of input for our authoring system. The users have to write scripts in ASL in which they list all the parts that they want to show in the video and outline the directions for the camera movements and animation of the elements seen. Our system reads the script, and the resulting animation is played in a Unity player. In the final part of the thesis, we present the results of our evaluation of the software done with four anatomy professors. The teachers could make their own narrated, animated video lessons on the knee and their limited experience in animation or video editing did not hold them back. The choice of text as an input was particularly favourable as the teachers did not have to learn a new and often intimidating user interface for design software, making our system intuitive and easy to learn.

Résumé

L'anatomie est une matière essentielle dans les études de santé mais aussi l'une des plus difficiles à enseigner. Elle fait partie des bases fondamentales, enseignées en début de cursus, qui sont utiles tout au long de la formation et dans la pratique professionnelle. L'un des défis majeurs de l'enseignement de l'anatomie consiste à transmettre des connaissances complexes en trois dimensions. Les méthodes multimédias telles que les vidéos animées et les modèles anatomiques en 3D sont très utilisées pour visualiser le corps humain. Mais la question du développement d'un logiciel facile à utiliser qui permettrait aux enseignants de créer leurs propres leçons sans avoir besoin d'experts en conception cinématographique et en 3D est toujours d'actualité. Dans cette thèse, nous présentons un système original de création d'un film pédagogique d'anatomie à partir d'un texte saisi par les professeurs d'anatomie. Nous fournissons la solution en trois parties. Nous présentons d'abord un langage formel pour décrire tous les éléments visuels d'une vidéo. Ce langage est à la fois lisible par l'homme et par la machine et il nous permet d'annoter la vidéo sous forme de texte. Ce langage Prose Storyboard Language (PSL) est développé pour une utilisation plus large dans la réalisation et l'analyse de films. Dans la deuxième partie de la thèse, nous présentons la spécialisation anatomique de ce langage appelée Anatomy Storyboard Language (ASL). Ce langage spécifique au domaine constitue la base de l'entrée de notre système de création. Les utilisateurs doivent écrire des scripts en ASL dans lesquels ils énumèrent toutes les parties qu'ils veulent montrer dans la vidéo et indiquent les directions pour les mouvements de caméra et l'animation des éléments vus. Notre système lit le script et l'animation qui en résulte est jouée dans un lecteur Unity (scène 3D). Dans la dernière partie de la thèse, nous présentons les résultats de notre évaluation du logiciel réalisée avec quatre professeurs d'anatomie. Les professeurs

ont pu réaliser leurs propres leçons vidéo animées et narrées sur le genou et leur expérience limitée en matière d'animation ou de montage vidéo ne les a pas freinés. Le choix du texte, comme entrée, a été particulièrement adapté car les professeurs n'ont pas eu à apprendre une nouvelle interface utilisateur, souvent intimidante, pour un logiciel de conception, ce qui rend notre système intuitif et facile à apprendre.

Chapter 1

Introduction

Anatomy is the cornerstone of medicine. It is broadly divided into two parts, gross anatomy or macroscopic anatomy and histology or microscopic anatomy. Gross anatomy deals with all the human body structures that can be studied with the naked eye. This includes the surface anatomy and all the internal organ systems. Microscopic anatomy is the study of cells and tissues that make up the organism and is studied using different microscopy techniques. Anatomy is one of the first subjects to be taught to students in the medical and paramedical fields. It forms the basis for understanding the healthy human body. From this foundation, the students are then introduced to the pathologies and trauma that can affect the body and the medical interventions implemented to treat these ailments. Therefore, healthcare professionals must have a broad knowledge of the human body and should retain this knowledge throughout their career.

Generally, the introduction to anatomical courses starts with lessons in gross anatomy. The students need a clear idea about the basic structure and relations of macro-level organs. After this, they are taught microscopic anatomy. This approach gives the student an overview first and then allows them to focus on different levels of magnification of that system under study.

The gold standard of anatomy education, over centuries, has been the dissection of cadavers. It is reflected in the etymology of the word anatomy from Greek, with ‘ana’ meaning ‘up’ and ‘tomia’ meaning ‘cutting’. Traditionally, lessons in anatomy

are divided into two parts. The first lessons are given didactically in a classroom where a large group of students listen and take notes from a lecture series given by an anatomy professor. The tools the professor use in these classes could vary from chalkboard drawings to pre dissected anatomical specimens. Still, in any case, the involvement of the student is limited to passive listening. In the second part of the lessons, smaller groups of students perform dissections on cadavers under the supervision of a professor. These sessions are more Socratic, and there is a discussion between the teacher and the students where the teacher provides not just information but also clinical scenarios relevant to the part being studied. The students are expected to be involved in dissection and the discussions. It ensures that they learn anatomy within the real-world clinical context and efficiently integrate theoretical and practical knowledge. This is the current system of instruction for anatomy. Still, with increasing class sizes, less number of anatomy professors, and reduced time allotted to anatomy studies in the medical curriculum, the necessity to augment this method with modern approaches becomes imminent.

With the increasing use of multimedia and 3D content in anatomy teaching, the issue now is the authoring of this content. Anatomy professors create all the teaching materials used in lessons, either by themselves or by others following their close instructions. In France, until a few years ago, the teachers would draw the part they were teaching on the chalkboard in front of the class as they led the lesson. The main reason for teachers to create their own material for the course is the learning objective. A learning objective is a clear and precise goal set before the lesson by the instructors or institutions for the students. At the end of a lesson, a student should successfully reach these objectives. For example, knowledge acquisition or a cognitive learning objective for a class on a bone could be that at the end of the lesson, a student must identify and name all the parts of that bone. The teacher then designs their lesson and course material to help students reach this objective. This further requires the teacher to have information about the learning objective the students have already achieved before the current lessons and that of the lessons that will be taught in the future. An added difficulty in anatomy learning is that specific fields require specific anatomical expertise. For example, medical students study anatomy from a clinical and surgical perspective, whereas physiotherapy students need to have high competencies in functional anatomy. Therefore teachers design anatomy lessons

with specific learning objectives. So, in essence, teachers need the creative freedom of an authoring system that is comprehensive yet straightforward to create and edit their pedagogical content.

The current anatomical content creation pipeline offers limited control to the anatomy teachers. Creating specialised content that meets the requirements of individual classes has not yet been resolved. In the current pipeline, if the anatomy teachers choose to incorporate 3D models and animated videos in their lessons, they either have to use the content already available or invest resources to create new content with a graphic designer's help. In the first case, the content may not match the class's learning objectives, and the second case offers very little control to the teachers over the finished video. Anatomy experts have the knowledge and teaching skills but do not have the training and tools to illustrate and animate the lessons. They have to hire graphic artists to make the videos under their direction. The artist takes the script given by the teacher and graphically interprets it in the movie. It is time-consuming and expensive, and there is a high chance that the learning objective may likely be lost during this interpretation. Since editing this content will require a repetition of the same laborious process, the outcome may not be what the teacher anticipated and they don't have the option to edit it later.

We aim to develop authoring systems that can be used to create animated, narrated videos for anatomy lessons. Our authoring systems do not require expertise in fields like cinematography, animation, camera operation or coding. Our goal is to have fast production, quick visualisation and easier editing giving teachers more freedom at different stages of content creation. We chose text as the method for input in our system as this fits seamlessly with the current way teachers create their lessons. Either for their live lecture series or when communicating with a designer to create multimedia content, it is common practice for teachers to write scripts detailing the concepts they want to teach. So, we make a text-based input system in which the teachers write scripts that describe all that they want to show in their final videos.

This work is done with the An@tomy2020 project that is funded by ANR(Agence Nationale de la Recherche). The aim of this project is to develop innovative tools to learn functional anatomy. It integrates the latest developments in graphics, human-machine interactions, modelling, cognitive sciences and pedagogy to make anatomy

learning immersive, effective and engaging. The main goal of the project is to promote embodied learning. It is a principle that stipulates that cognition is linked not just to the mental faculties but also to the physical body and gestures of the subject. The subject's interactions with their environment affects their understanding capacity. This is especially prominent in anatomy as the learner can better visualise and comprehend complex anatomical concepts when they apply them to their own body. An@tomy2020 aims to facilitate this 'embodiment' by letting the students visualise their anatomy using augmented reality. The project has three parts. The first aims to create a robust motion capture system to record the learner's movement and then project the animation onto their own bodies. The second part aims to develop interactive strategies in augmented reality. The third axis of the project focuses on developing educational content and scenarios for anatomy lessons, creating an authoring tool that teachers will use to produce this content and evaluate the efficacy of both the authoring tool and the overall contribution of An@tomy2020 in the improvement of learning. This thesis is part of the project's third axis and describes the authoring system's development and evaluation.

The first step to understanding anything would be to understand and identify the individual parts that make up the whole. To this end, we describe a formal domain-specific language called the Prose Storyboard Language to describe all the essential visual elements that occur on screen that make up the movie. It is a simple, human and machine-readable language that integrates the semi-formal idiomatic vocabulary already used in movie making. Prose storyboard language forms the basis of the text-based input methods for all our authoring, editing and annotating systems.

The use of a specialised form of natural language is prevalent in traditional movie-making. Directors use it to convey their ideas for shots to their crew, especially the cinematography department. It is further used in film studies and by critics to annotate the scenes to understand them better. As we move into the world of virtual cinematography and content creation by non-expert users, the need to formalise the descriptive cinematographic language becomes paramount. The goal is to develop a language with a gradual learning curve for new users while being comprehensive enough to describe the movie. This description can then be used as an input by cinematographic software to create fully realised shots. In essence, Prose Storyboard

Language details all the agents seen on the screen (compositions) and tracks all the actions over time that lead to changes in these compositions.

Prose Storyboard Language is a comprehensive and extensive language that can faithfully describe the visual elements seen on screen. For the use case of anatomical teaching videos, we need to simplify and specialise PSL to suit the needs. For starters, the anatomical parts are all interconnected and the movement of one will affect the movement of the other. Hence, an individual description of actions for each actor, as is done in PSL, is not necessary for its anatomical variant. There is also no need for stylistic transitions and camera movements in a pedagogical scenario. Therefore, we tailor PSL to an anatomical teaching perspective to create the Anatomy Storyboard Language or ASL. Scripts written in ASL are the input for the authoring system. They describe the anatomical parts seen on screen, the camera movements around the parts and the actions that these parts perform. Using this information, we can build and animate the 3D scenarios. The ASL scripts are translated into hierarchical Finite State Machines. These state machines are read by the Unity application developed by our collaborators in Anatoscope to generate an animated video.

We then add the feature for having narrated voiceovers for the video lessons. It enables the teachers to create a complete narrated, animated video. The teachers write the visually descriptive script for the video in ASL and then record their voiceovers. Our system combines these two and synchronises the audio from the narration to the video.

After we had the authoring system, four anatomy professors evaluated its usability. In the evaluation sessions, we first determined the need for such an authoring tool, the extent of use multimedia in their current lessons and the teacher's prior experience with digital content creation. Then we gave them a tutorial on ASL, good practices for scriptwriting in ASL and tips to record their narrations. They were encouraged to build their first 3D scenes using ASL and explore its creative possibilities. After they were familiar with the system, they created four short lessons on the knee's anatomy with our assistance. At the end of the authoring sessions we administered the NASA Taskload test to identify the source of the mental workload for our authoring system. Despite this being the first system the professors used to create videos by themselves,

they did not report a very high cognitive load. Instead, they were enthusiastic about using our system in future to create videos for their classes.

Finally, we propose an extension of the authoring system to create interactive lessons. Interactive pedagogy has shown great potential in helping physicians prepare for patient interactions and patient education. Even in anatomy education, many applications and websites allow students to interact with 3D models of the human body. Students have reported using these models beyond their regular courses to better understand the spatial relations between anatomical parts. We would like to formally combine this interactive experience with the structured anatomy curriculum. We believe the way to achieve this is to develop authoring tools that teachers can use to write their own gamified lessons. With this, we give them a chance to tailor-make interactive content for their students based on learning objectives.

Chapter 2

State of the art

This chapter presents the state of the art research done in teaching anatomy. We present the current methods of teaching anatomy and the subsequent rise in the popularity of mixed media content. Systems for anatomical content creation lead us to the broader field video creation and raise the necessity to formalise a language to describe visual content. We discuss the different software developed to control the camera in a virtual space and text inputs for these systems. Next, we highlight work done in natural language processing for descriptive cinematographic languages. We then present semantic models used to describe videos, games and other media creation.

2.1. Approaches to anatomy teaching

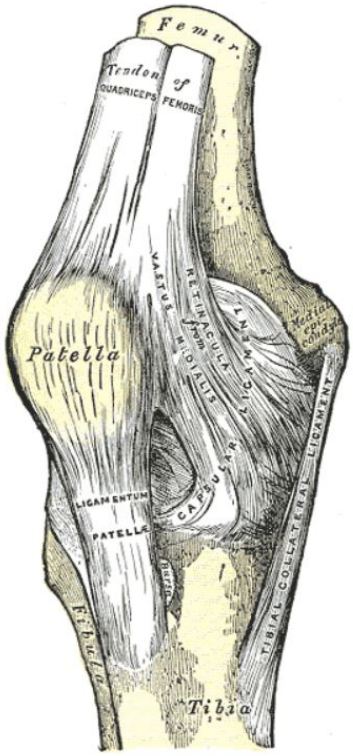
Anatomy learning has two parts: theory and practical. The lesson starts with classroom lectures with visual aids such as chalkboard drawings and slide presentations. These sessions are for the entire class, and then students are divided into smaller groups for practical dissection sessions on cadavers and prosections. The students have better access to the teachers to ask questions and are informally evaluated on their knowledge by oral questioning during the practical sessions. Some of these teaching references are presented in the figure 2.1. However, due to increasing class sizes, reduced faculty, and less time allotted for the anatomy curriculum, augmenting traditional

teaching methods with multimedia approaches has become imperative. All anatomy professors have been looking for new solutions to these existing problems. In our survey, presented further in the thesis, they have stated that they are willing to invest their time and expertise to develop digital content by themselves. Furthermore, in light of the current Covid19 pandemic and subsequent measures of social distancing and remote working, it has become vital to come up with creative solutions for distance learning, especially for complicated medical subjects [75]. In this thesis, we present innovative solutions to meet these pre-existing issues.

Various multidisciplinary techniques have been introduced to make anatomical learning more engaging and effective [32]. [10, 117] discusses the role of dissections as the gold standard in anatomy classes. While there cannot be a complete digitalisation of anatomical lectures, it is beneficial to use multimedia tools as a part of the blended learning approach to better prepare the students for the hands-on practical sessions [63, 108, 54]. This approach is not only engaging for the students but also cost-effective [54, 26]. As anatomy involves the study of complex spatial structures, it can benefit from recent results in computer graphics and informatics [114]. These resources make it possible to produce animated videos with camera actions and physiological movements that enable better visualisations of the human body. While developing new content, it is also essential to control its quality and efficacy in real-life scenarios. [116] outlines the need for the most efficient and appropriate use of multimedia in anatomy courses from the perspective of a student's cognitive load.

The effectiveness of new methods has been confirmed in previous studies [47, 84, 113]. Pereira et al. use a mix of teacher-created recordings and previously available material, whereas Hoyek et al. use the extensive library of 3D animated videos created by Lyon 1 university in collaboration with a graphics team (figure 2.2) [1]. Both these studies highlight the need for a teacher authoring system for animated content creation. The videos made by the Anatomie 3D Lyon team have been integral in our process of designing and developing an authoring system. As these videos fit the standard practices of teaching anatomy in France we modelled out final output to be similar to them. Animated videos are the most commonly used teaching aids in anatomy [48].

[1] https://www.youtube.com/channel/UCHKlhyLLAxFA69nO0NK9wPA



(a)

(b)

(c)

Figure 2.1: Anatomical teaching references (a) Illustration of Right knee from Gray's Anatomy 20th ed., (b) 3D model of Right knee from Zygote model of human body, (c) Functional Right knee joint model from Amazon

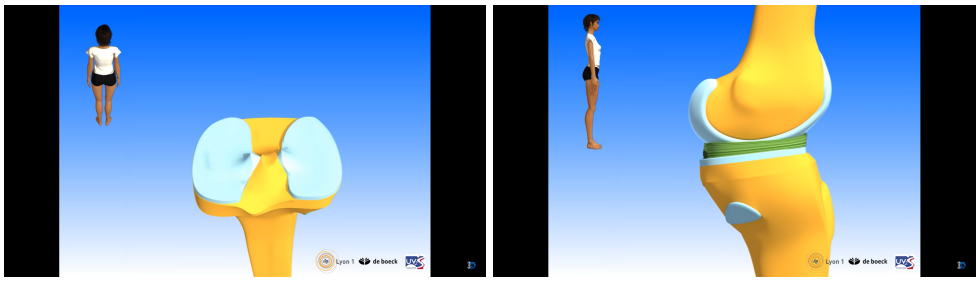


Figure 2.2: Stills from "Articulation du genou" video made by Anatomie 3D Lyon

Even beyond the course suggestions, students reported using video sharing sites such as YouTube to learn new concepts, clear doubts or better visualise anatomy [50].

In recent years more advance and immersive techniques are being used to increase embodiment in the learning process. A study by Kucuk et al.[56] shows that students using mobile augmented reality (mAR) technology to learn anatomy had higher achievement as they were able to formalise abstract anatomical data. With the development of augmented reality, [13, 51] and virtual reality [78, 107] in anatomy learning, an easy-to-use text-to-movie tool is imperative.

Authoring high-quality 3D animation is a costly process requiring the skills of designers, animators and directors. Likewise, creating 3D animation for a new course in anatomy involves time, effort and money. Few authoring systems are easy enough for an anatomy teacher not trained in computer graphics to create 3D animation by herself. Recently, commercial game engines have started providing visual programming tools to facilitate the creation of 3D animation (Unreal Engine's blueprints [2], Unity's Cinemachine [3] and Timeline Editor, Huttong Games' PlayMaker [4]) and even spatial programming tools to facilitate the creation of mixed-reality content (Unity's EditorXR [5]). But they are better suited for expert game developers than medical anatomy teachers.

[2] https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/

[3] https://unity.com/unity/features/editor/art-and-design/cinemachine

[4] https://hutonggames.com/

[5] https://github.com/Unity-Technologies/EditorXR

As a result, teachers are forced to buy existing 3D animation from commercial companies or to work with professional animation studios to create novel content. Creating mixed reality applications is even harder, requiring the skills of real-time video game designers, artists and developers. Clearly, there is a need for authoring tools providing better support for authors who are experts in their own field (human anatomy) and not in computer graphics and animation.

2.2. Text-to-movie authoring

This thesis aims to take a text-based input, like a script and create a video from it in the anatomical lesson use case. In essence, we want to codify the mental process of reading a script and imagining a developing scenario. To do this, we first looked at formalising the language for these scripts. We start with the broader field of movie-making.

The first step in creating a video or a movie is writing the script. The script is a complete description of all things that take place on the screen. It should be dynamic with many details regarding the events that unfold in the run time of the video. It should describe the compositions, actions, and stage directions so that the reader can recreate the scenario in their head based solely on this text, sort of like a prose storyboard. A storyboard is a graphical organisation of the script with images and illustrations. It is used mainly in the pre-visualization step of filmmaking, where the director can visualise the script and plan the movie before it is shot. They can block the characters, experiment with the compositions, and fine-tune the camera choreography [9]. This allows creators to see the film from the camera's perspective and saves time and money for the principal photography or the filming stage. A prose storyboard is a written description of the action and directions.

The term prose storyboard was first used by Proferes [89] in his book "Film Directing Fundamentals - See your film before shooting it" to break down a script into it is component shots. He used natural language to describe this prose storyboard. Natural language is the everyday language used by people for communication. It evolves naturally over widespread use. In contrast to Proferes, our system uses a

formal language with a well-defined syntax and semantics, suitable for future work in intelligent cinematography and editing.

Previous work on establishing a formal grammar for a shot in the movies has been described in the works of Thompson and Bowen in [111, 112] and Arijon [4]. The current vocabulary to describe shots also draws heavily from the works analysing film style [14, 98]. These give us the terms necessary to build our specialised descriptive language for movies.

Our proposal is complementary to the Movie Script Markup Language (MSML) [92], which gives a structured format to screenplays and movie scripts which helps in the production and future automation of content creation, given the screenplay. It has four models. First is the Scene Model, which consists of the narrative part of the content and has the standard structure of a screenplay. It breaks down the components of the narrative and provides the identification and description for all the entities and events involved. These entities can either be characters or props from the story. Events push the narrative forward by affecting the entities in the scene, either through actions or dialogues. The Scene Model describes the entities and events within the narrative but does not include the instructions necessary to produce the story. These instructions, such as camera placement and movements, and stage and lighting, are not part of the narrative but are essential to creating the scene. They are included in the Manufacturing Model. A combination of the Scene and Manufacturing Models provides all the necessary information to set up entities, play out the events, and direct the camera and stage setting. To accurately represent the semantic of the scene, information about the timing of all significant actions is also necessary. This is provided in the MSML Timing Model. It ensures that synchronisation between events can occur at any moment within the event's execution. This means two events can be linked at any arbitrary moment in time of their execution. The model also ensures events can get linked such that the occurrence of one event affects the other. Furthermore, the duration of each event will also affect the relationship between them. This fine-grained approach to timing and synchronisation allows the creation of complex dramatic scenes and also help in future automation of dramatic production. The final part of MSML is its Animation Model, which is an addition to the traditional screenwriting tools. As the earlier three models provide all the information to set up a

scene and play the events with detailed information about timing and synchronisation, it is possible, in theory, to use this information to construct and play the scene virtually in the Animation model. Our work is similar to MSML, which breaks the script into dialogue and action blocks using its four models. But MSML does not provide a hierarchy of a movie or describe the actual translation of these blocks into movies. It does not represent the movie as a series of shots as we do. We also elaborated on their Animation Model by building a system that plays the screenplays or scripts written in our domain-specific language. We illustrate this with examples in anatomy teaching.

Another important work is the Declarative Camera Control Language (DCCL) which also describes film idioms, not in terms of cameras in world coordinates but in terms of shots in screen coordinates, [24]. The DCCL is compiled into a film tree containing all the possible editings of the input actions, where actions are represented as subject-verb-object triples. Our proposed language can be used in coordination with such constructs to guide a more extensive set of shot categories, including complex and developing shots.

Other previous works in virtual cinematography [101, 52, 37, 80, 64, 38, 39, 57] has been limited to simple shots with either a static camera or a single uniform camera movement. The Prose Storyboard Language is complementary to such previous work. It can be used to define higher-level cinematic strategies, including arbitrarily complex combinations of camera and actor movements, for most existing virtual cinematography systems.

Text-to-movie authoring is a general class of methods that have been proposed for automatically generating 3D graphics and animation from text written by a domain expert. Xtranormal Technology Inc. first used the term text-to-movie to describe their authoring system that combined the previously used text-to-speech and text-to-scene concepts. Text-to speech enables text to be converted into speech signals that imitate human voice and intonation. This can be used to convert written dialogue in scripts into speech or as voiceovers in instructional videos. Text-to-scene enables the visualisation of natural language descriptions. This needs 3D models that are labelled and positioned in the 3D world depending on the text descriptions. A combination of these two technologies, along with the addition of animations and interactions of virtual actors with the 3D scene based on text descriptions, falls under the domain of

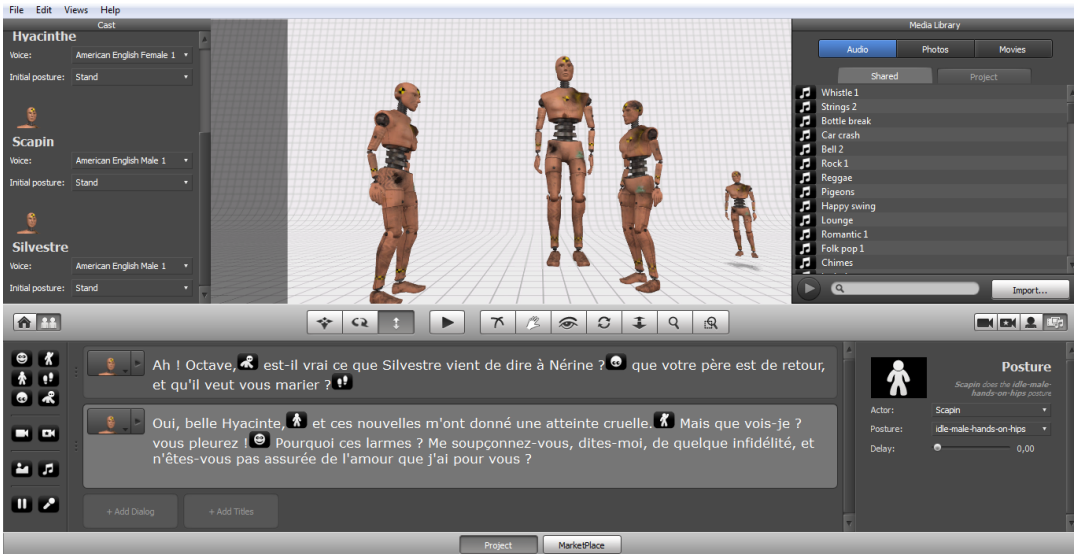


Figure 2.3: Still from the Xtranormal text-to-movie authoring system: STATE. The text written for each actor is the dialogue and the markers seen on the lower left part of the screen direct the camera and actor actions within the scene

text-to-movie authoring. An example of this is seen in figure 2.3. It is from a system called State developed by Xtranormal in which the "text" consists of dialogues and a combination of markers that are similar to emojis. These markers direct the actor movement, facial expressions, posture, interactions and also the camera movements. In essence, the input method is a mix of natural language text and markers that direct actions, both camera and actor. The output can be visualised immediately as the 3D scene is being built and directed.

Good results have been obtained in limited domains, such as generating 3D scenes from natural language accident reports [1, 79] or generating cartoon animation from scripted dialogue scenes [100]. Commercially available text-to-movie systems such as NawmalMAKE (from Xtranormal Technology Inc) [6] and Plotagon Studio [7] are specifically designed to generate dialogue scenes in selected cartoon styles. The Xtranormal text-to-video software has been used in student-centred earning approaches [109]. In this study, they ask the students to use the text-to-video feature of Xtranormal to create short training sessions in which they played the role of

[6] https://www.nawmal.com/
[7] https://plotagon.com/

Human Resource Management trainers in a hospital. In the scenario, students were responsible for orientation new hires and created training videos for them. The study found that by giving higher creative freedom to the students to author their videos, they engaged better with their course content and learned the material in greater depth to create better videos. The students also had a better understanding of the real-world application of their knowledge because of the increased engagement.

Generic text-to-movie authoring is also an active area of research. Ye and Baldwin described a system for automatically generating storyboards from natural language movie scripts [123].They present a machine learning-based Natural Language Processing system to produce animated storyboards based on the action described in the script. They do this by identifying the verbs and their semantics in the script and applying that to the virtual stage. The virtual stage is built by using a drag and drop interface and has annotated 3D models with inbuilt real-world knowledge of the objects they represent and a system to query their status, such as their position and orientation in the virtual world at any time. The main task here is to identify the timeline and the context in which the verbs are used to describe the action in the natural language script. This, combined with the information from the virtual stage, is used to create an animated storyboard. We simplify this issue by using a formalised language. In our system, the verbs are listed in the grammar and this list can be expanded as needed by the user. An animation library can be built for the actions listed, and when the user writes the script using our language, the actions' animations can automatically be played.

Storyboards themselves have also been used as input methods to craft stories. CANVAS [53] is a visual authoring tool in which users describe key plot points in a narrative with multiple characters as logical interactions between them. This is done in the form of visual storyboards. After the user defines the story as best as they can, CANVAS fills in the gaps in the narration using logical actions and interactions between the characters and generates a completed story while preserving the original intent of the creator/user/author. This story can then be visualised as an animation. This instant pre-visualisation allows for multiple iterations and rapid prototyping of different narratives. CARDINAL [67] is a natural language-based script authoring system that allows interaction and previsualization of script elements, but the focus of CARDINAL

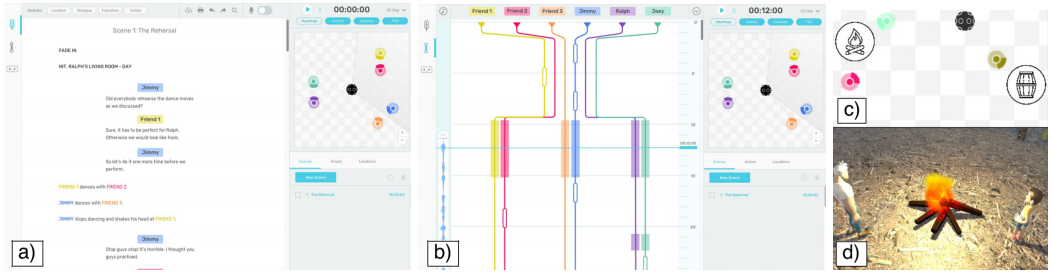


Figure 2.4: Different views from Marti et al.'s CARDINAL system. a) shows the Script View b) shows the Interaction View where the author can visualise how the characters in the story interact with each other and c), d) show the 2D and 3D pre-visualization of the script respectively.

is more on mapping character interactions rather than directing the story. Details of the CARDINAL system are seen in figure2.4. This is also seen in the work of Won et al.[119] in which they create pre-visualisations of character interactions based on high-level text-based descriptions. They generate and then rank the animations that best suit the sparse description of the action. Beyond authoring, storyboards have also been used as points of reference to manage the production of a movie. Bartindale et al. [8] present a prototype called StoryCrate, which is a tabletop collaborative system that uses storyboards to track the progress of the shoot among different departments.

In recent years, research in the use of neural networks to create videos based on text input is being developed [68, 43]. Generative Adversarial Networks (GANs) have been used to synthesise images. Pan et al. [83] to use GANs to generate short videos from captions. This is non-trivial as videos are a sequence of temporal and semantically linked images and to correctly translate a text-based input to synthesise a video is challenging. Text-based editing is also gaining popularity. Fried et al. [36] present their work on editing talking-head videos (an actor speaking with the camera focusing on their face and upper body) by making changes in the transcript.

Closer to our approach, Director Notation [122] is a symbolic language intended to express the content of the film (motion pictures), much as musical notation provides a language for the writing of music. But DN is a graphical notation, whereas PSL is a pseudo-natural language, and DN describes the movie production process, whereas PSL describes the movie itself, as in a storyboard. TIMISTO [115] and SLAP [15] are

pattern languages for creating animation from storyboards. Finally, the video language [2, 34] is a special-purpose language for scripting repetitive tasks in video editing written in the Racket programming language [35]. The video language operates at the level of video files and frames and does not include a description of the shot content, which we are targeting here.

The first Prose Storyboard Language formalisation of visual elements of a movie in a text format was developed by Ronfard *et. al* in 2015 [93]. In this version, they introduced a domain-specific language used as a high-level user interface to create and annotate movies. It contained grammar to label the different parts of a shot. In these first steps of formalisation, the definition of a shot was highly simplified to a combination of composition and screen events and screen events were further divided into events that changed the composition and events that did not. While this is correct, it simplifies the language to such an extent that it becomes difficult to describe complicated shots with a lot of activity after the initial composition. The complexity of the choreography of camera movements and actor actions are lost in such a simplification. Over the course of this thesis, PSL was expanded and restructured. We introduced a new concept of *developments* that included all events that took place after the initial composition. This development was further divided hierarchically into *continuations* and *recompositions*. This partonomic and taxonomic hierarchy of a shot was missing in the initial versions. We also modified the syntax to reflect the new hierarchy, introduced a parser to break down the PSL sentence into its parts and developed a schema for the semantic translation of a PSL sentence into a graphical Petri net structure. We then tested this version by annotating four long shots with various complexities and many short examples of different shots. Using the new version of PSL, we faithfully transcribe all the visual elements in these examples. This is described in further detail in chapter 3. This version was then used as a basis to develop a specialised language for anatomy video authoring called the Anatomy Storyboard Language. Several variations of PSL have also been used for generating cinematic replays in serious games [38], for generating synthetic complex shots from live video material [42, 41], for staging complex scenes in 3D animation [61], for directing cinematographic drones [40] and for learning film editing patterns from examples [120].

2.3. Real-time animation

The next step of the authoring process is to create animations from the text-based scripts. A way to achieve this is to convert the scripts into models that can be read by game engines such as Unity in real-time. These models need to have all the descriptive information of the scene and the details of the events and their timing. These semantic models help us create meaningful Prose Storyboard Language descriptions that are syntactically correct and faithfully read by an algorithm to generate a video. Semantic checks, in our case, ensure that all parts of the sentence are correct and work together coherently. This means that the final video will not just be a collection of random visual elements. It will have the narrative flow that the user wants. The semantic modelling for PSL is a Timed Petri Net (TPN). They are a variant of the Petri nets that were introduced in Carl Adam Petri's PhD dissertation in 1962 [86] to represent parallel actions in a system. Petri nets are mathematical formalisations that model the directional flow of information. They are directed graphs that model discrete events in a system. They describe changes in a system where the states or conditions are shown as places, and the actions required to move from one state to the other are shown as transitions. In the graphical notation, places are drawn as circles or ovals and transitions as squares, rectangles or lines as seen in figure 2.5. Directional arcs connect places and transitions. Arcs are only allowed to connect a place to a transition or vice versa. Two places or two transitions are not allowed to interconnect. Places contain information about the resources that need to be attained or the conditions needed to be met for an action to occur. An action (denoted as a transition) can only happen when all the conditions or states that lead to it are fulfiled or true. When this transition is fired or executed, it allows the system to proceed to all the places connected immediately downstream.

Aside from places, transitions and arcs, Petri nets may also contain tokens. Tokens are a representation of conditions at a place. The distribution of tokens in different places of the net gives a snapshot of the system's state. It is referred to as its marking. Only places can hold tokens. Arcs can carry weights or specifications such as, in simple cases, the number or type tokens they can carry. Tokens move from one place to another via transitions. As mentioned earlier, a transition is enabled when it fulfils all the requirements of its input places (all the arc that leads to the transition must carry

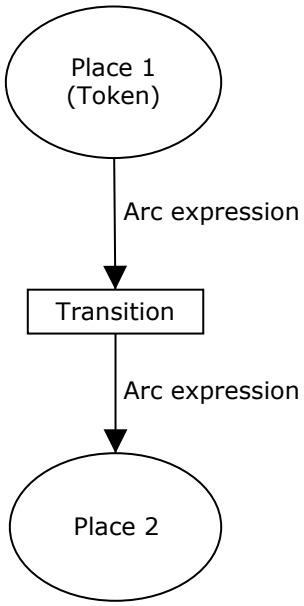


Figure 2.5: Graphical representation of a Petri net

the tokens corresponding to their arc weight). When an enabled transition is fired, it consumes the tokens from its input places and generates them in the output places according to arcs specifications and place requirements. It is important to note that transitions don't simply pass tokens from one place to another when fired. Instead, they consume and generate tokens.

Petri nets can range from modelling industrial processes [125], human-computer interactions [91, 29] to pedagogical fields such as collaborative learning [49, 59] or developing personalised learning tools [27]. They are instrumental as they can model expansive data sets with concurrent or synchronous actions.

The Petri net model works well to model PSL as movies and videos are directional (they follow the linear timeline of the video), and multiple actions take place simultaneously on the screen, independent of each other. This free agency and concurrency of actor and camera movements can be well represented with Petri nets. They have also been used to model systems in other arts and media such as music and gaming.

Petri Net music models have been proposed to represent music at an abstract level suitable for analysis and generation of music by computers. Scoresynth is an early

example [45]. In recent work, Barate et al. have used the generic term "Music Petri Net" to describe Petri Nets whose places are music objects (notes, beats, chords, etc.) and whose transitions are music transformations (transposition, harmonization, etc.) [6]. [95] describe a method to create Petri net models that can compose music by building on its basic components such as vocal and rhythm scales.

Similar to the case of Music Petri Nets, we can think of Game Petri Nets as a general class of Petri nets whose places are game objects. Taking film analysis as a foundation, Natkin and Vega attempt to provide a formal representation for storytelling based games [77]. They propose a Petri Net specification illustrated on the well-known game Myst. In their paper, Barreto and Julia [7] present a system of designing video games. They model the activities and interactions the players can have in the game in one net structure and the map of the virtual world with key topological areas in another net. Then they propose a third net that shows how the activity model interacts with the topological model. Lee and Cho [58] propose using Petri nets to procedurally generate complex game plots for Role Playing Games to keep with the high demand. Another approach uses Petri nets to create nonlinear plots and manages story progression in virtual worlds [16]. Beyond modelling game mechanics, Petri nets have also been used to map the production framework for animation and video game industries [71].

In general, the firing of transitions in a Petri net is instantaneous and occur when all the input requirements are fulfilled. We can extract the time at which these transitions are fired, but unless specified, this is the simulation time calculated by the internal clock within the net. It does not translate into real-time; it only indicates the order of firing of transitions in the system. In the case of videos where each action takes place at a specific time and actions have a duration of time, we need Petri nets that accurately model this. This can be done using Timed Petri Nets or TPNs. [103] provides a review of the different types of Timed Petri nets. Other fields that use TPNs are chemical manufacturing [17], modelling safety requirements at train crossings [31] and scheduling of assembly tasks between different agents [18]. All these use cases use real-world time data in the net to fire transitions. As mentioned earlier TPNs have also been proposed for representing the temporal structure of movie scripts [92], character animation [62, 12], game authoring [5], turn-taking in conversation [20] and synchronisation and storage models for multimedia systems [60]. This final

example by Little et al. [60] is especially important to our work as they combine the use of Timed Petri Nets and time annotated data to create content. It is a method of sequentialisation of temporarily related media objects using Petri Nets. They explain this further using their example case of an Anatomy and Physiology Instructor. It consists of a database of medical knowledge in slides, audio recordings, images and videos. These media objects have time annotations attached to them. The user can build their content in sequential order from this database. They can choose to show multiple media elements simultaneously or order them in a specific manner according to their requirements. The Timed Petri Net provides this synchronisation. It regulates the playing of the time-labelled media content. We use a similar process to model Petri Nets for Prose Storyboard Language and regulate how different media elements are played.

Chapter 3

Cinematographic language

This chapter describes the formalisation of visual elements in a video in terms of domain-specific language. This language is called the Prose Storyboard Language. It can be used throughout the movie production pipeline, starting from early prototyping and pre-production when the story is communicated between screenwriters and directors, to production, as a guide for filming and finally in post-production for precise editing. An example of the language in use is seen in figure 3.1

3.1. Structure of the Prose Storyboard Language

To formalise the visual elements seen in a video, we break down and categorise all the key visual elements into a hierarchical structure with partonomic and taxonomic groups. In this way, we can label the parts, describe their relationship to each other and codify a visual medium in descriptive text. This hierarchy is visually represented in an And/Or chart in figure 3.2. As mentioned, this hierarchy consists of partonomic and taxonomic classifications. Partonomy or meronomy is a type of hierarchy that describes a 'part of' relationship. It breaks down a whole object into its respective components. It is usually described from top-down with a 'has a' or 'made of' relationship. For example, a car has an engine and tyres. Taxonomy, on the other hand, is a classification that groups or creates sets of similar things. The relationship here is a 'type of' or a 'class of' relationship. For example, sedans and hatchbacks are types of cars. Partonomies

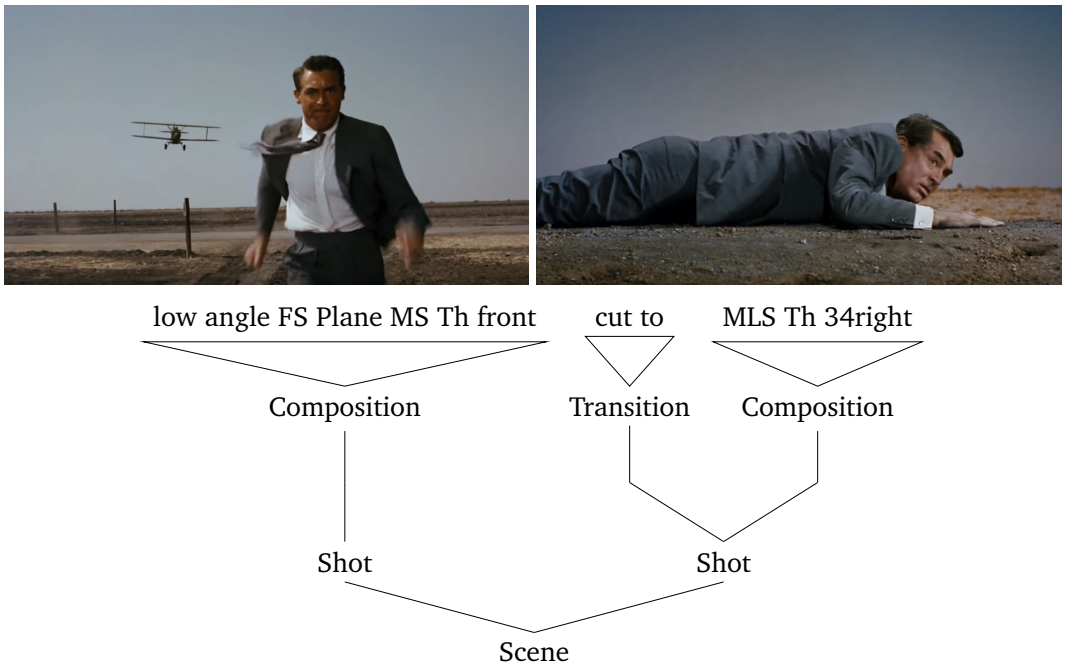


Figure 3.1: Prose storyboard language description of two iconic shots in Alfred Hitchcock's *North by Northwest*

and taxonomies help us understand the exact relationships between different elements in a system.

Applying these principles of classification to a video, we can say that a movie is made of a series of scenes. A scene is a part of a movie where the action occurs either in a single location or in continuous time or both. It is made of a series of shots. In the final product or movie, a shot is a continuous series of recorded frames with a camera. The definition varies slightly at different stages of production. In the pre-production stage, the shot is a planned series of frames in the script or the director's mind. In production, it is what the camera records between the director calling 'action' and 'cut'. In post-production, these raw shots are cut and assembled together to make the final shot by the editor. This descriptive language can describe shots at any of these levels as they maintain the same hierarchical structure.

A shot is made of 3 parts, *transition, composition* and *developments* (figure 3.3). Transitions describe how we enter a shot; compositions describe the visual elements in

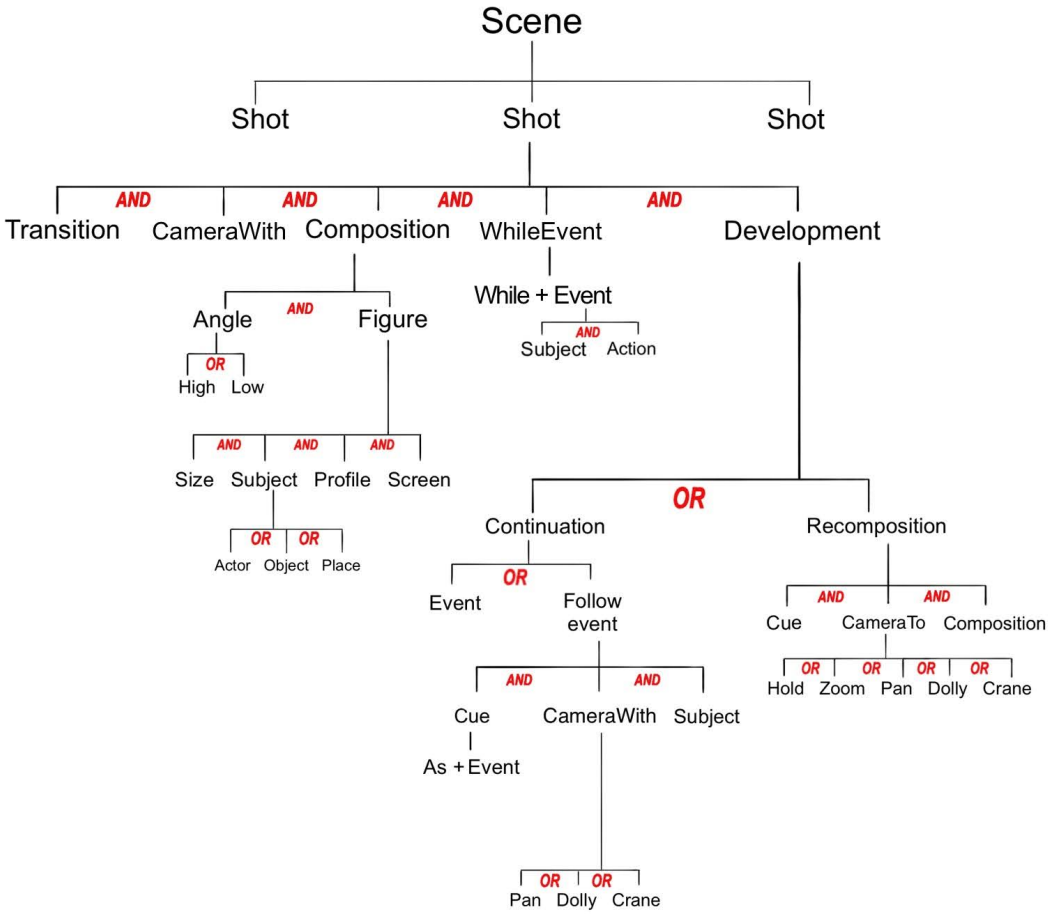


Figure 3.2: AND-OR tree representation of the Prose Storyboard Language grammar.

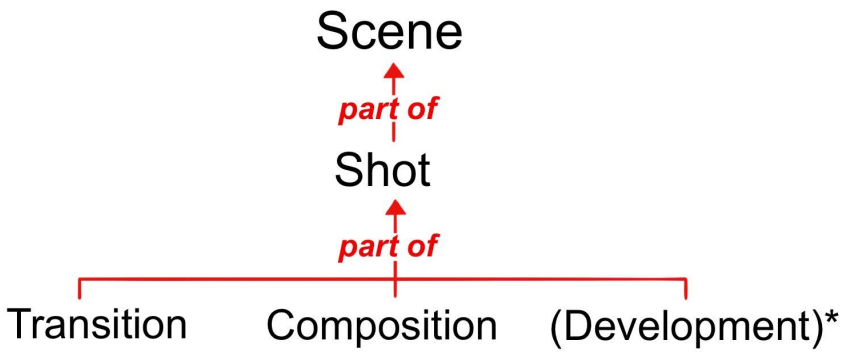


Figure 3.3: Partonomy of a Shot

keyframes, and developments describe how a shot progresses from one composition to another.

Transition specifies the way of progression of a shot into the subsequent one. In our model we include three of the most widely used transition techniques: *cut*, *dissolve* and *fade*. We use the simplest form of cut transition in which the two shots are played one after another. We also use the same notation to describe other types of cuts, such as cutaways in which shot A is followed by an intermittent shot with a different composition and then returns to shot A. Dissolves and fades are used to describe the entry or exit of a shot in which the composition either slowly appears or disappears respectively. Cut, dissolve and fade are types of transitions. In other words, they belong to the taxonomic group of transitions.

The second part of a shot is the composition. It is the first set of visual elements that we see when we enter the shot and shows a tableau of actors in a setting. The actors are not performing any actions, and the camera is stationary. The composition describes all the key visual elements in the first frame, such as the actors, their profiles, placement in the screen space and more.

Any change from the initial composition is written as *developments*. Developments are the third part of a shot. They are optional as there are shots where we transition into an initial composition and then transition into another shot without any changes in the visual arrangements to the initial composition. Developments are of two types, *continuations* or *recompositions*. We will now describe the hierarchy of partonomical groups of composition and developments, starting with the initial composition.

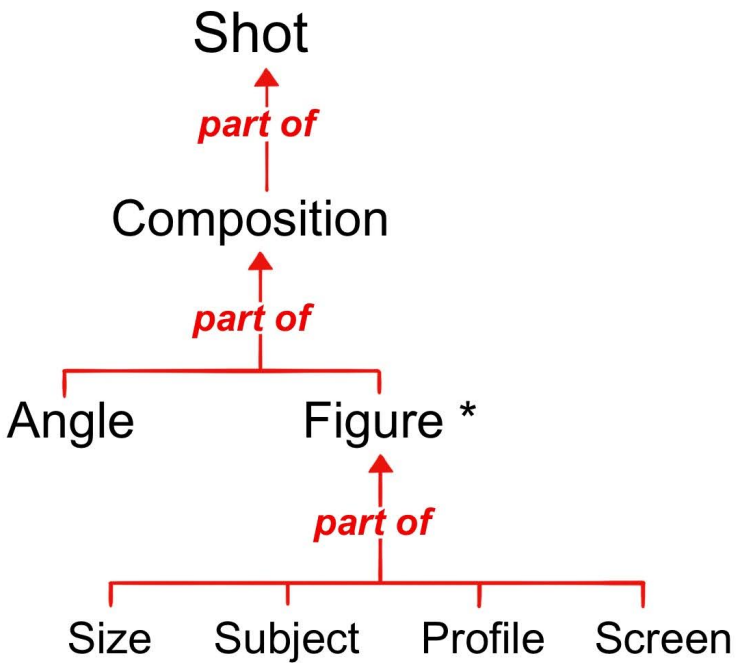


Figure 3.4: Partonomy of Composition

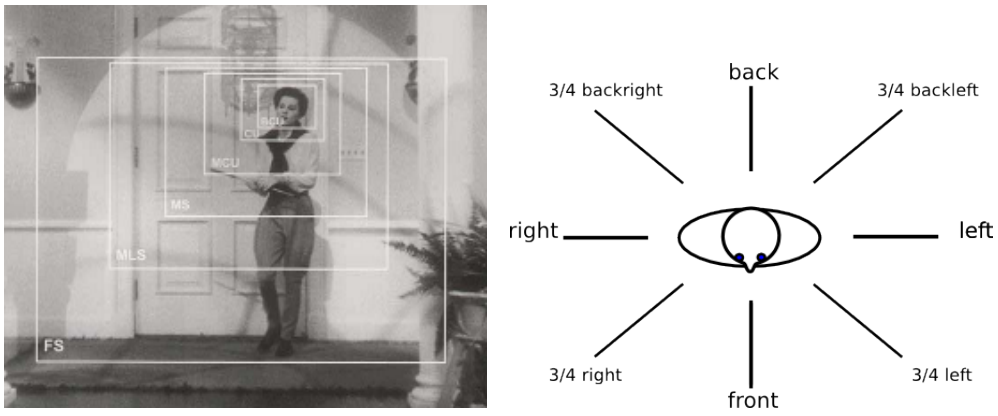
3.1.1. Image composition

Image composition is the way to organise visual elements in the motion picture frame to deliver a specific message to the audience. Our work proposes a formal way to describe the image composition in terms of the actors and objects present on the screen and the spatial and temporal relations between them. All these descriptions are made concerning the camera as that is what the viewer will finally see. Following Thomson and Bowen [111], we define composition as the relative position and orientation of visual elements called *Subject* in a frame. A composition has an *angle* and a *figure*. The angle specifies the position of the camera in relation to the subjects in the composition. The default is the straight, horizontal view, but there are two special types, the *high* and *low* angles. In a high angle view, the camera is above the horizontal plane of view of the actors, and it looks down upon the subjects from this elevated perspective. Consequently, in a low angle view, the camera is below the horizontal eye level of the actors and looks up at them. The angle is described before the individual actors are, as it is a property of the camera, and it stays consistent for all the actors in the composition.

The second part of the composition is the figure. It is made of four parts, *size*, *subject*, *profile* and *screen*. The size element describes the relative sizes of subjects in the composition. It depends on the distance at which each actor is standing concerning the camera, as illustrated in figure.3.6(a). In the simple case of *flat staging*, all subjects are more or less in the same plane with the same size, but in the case of *deep staging*, different subjects are seen at different sizes, in different planes. Furthermore, as a convention, we describe the subjects from left to right. It means that the left-to-right ordering of actors and objects is part of the composition. An illustration of both individual descriptions of size and the left to right ordering is seen in the figure 3.10(a) and 3.10(b). As Salt outlines in his book, Moving Into Pictures [97], we describe the sizes from an *extreme long shot* or *ELS* to an *extreme close up* or *ECU*. In the former, the camera is placed very far away from the subject, and we can see the whole of the subject and a broad view of the rest of the setting, and in the latter, the camera is focused very close to a part on the subject, and we see a very narrow view with that part in emphasis. All these levels of focus or magnification are types of sizes. They are shown in figure. 3.5(a) and the abbreviations are listed in table 3.4.

After we describe the sizes, we specify the subjects. Subjects are of three types, actor, object and place. The subject is the main focus of the composition. It is the part of the composition we describe using the other parts such as angle, size, profile and screen. In our language, there is no limit placed on the number of subjects that can be listed, so it can be used to describe frames with any number of actors and objects. Having established the subjects, we then specify the profile in which we are viewing them. The profile is defined as the part of the subject, generally the actor the camera sees. The visual representation of this is seen in the figure. 3.5(b).

The final part of the figure description is the screen. It specifies the subject position in the screen coordinate. It allows us to describe slightly modified framing to the generic central framing by shifting the subject position to the left or right corner, as shown in figure.3.6(b). Default screen values are used to describe symmetric compositions where subjects are evenly distributed from left to right. Non-default screen values are used to describe asymmetric compositions, e.g. taking into account head-room and look-room or the rules of thirds. We can also describe unconventional framing

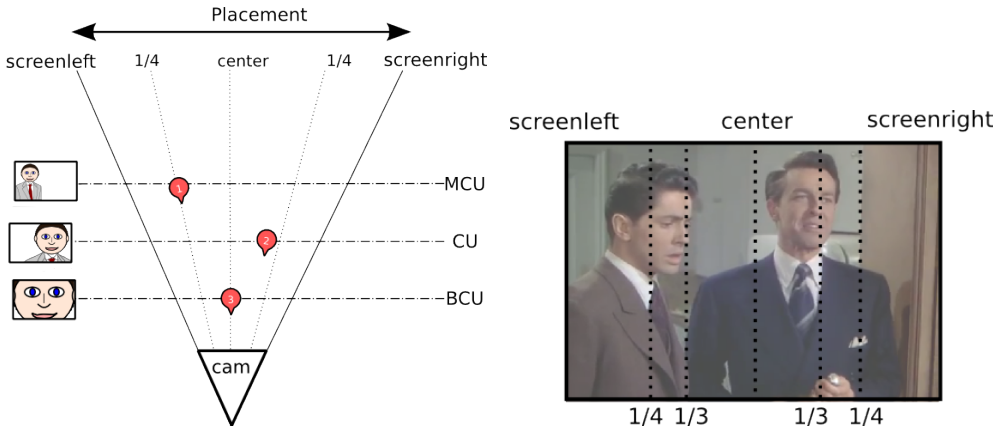


(a) Shot sizes (b) Profile angle

Figure 3.5: (a) shows shot sizes in the prose storyboard (reproduced from [97]). (b) shows the profile angle of an actor defines his orientation relative to the camera. For example, an actor with a *left* profile angle is oriented with his left side facing the camera.

to create an unbalanced artistic composition or to show other visual elements in the scene.

Above mentioned parts and types of the language are necessary to build the usual composition. In some instances, the shot can directly transition into action, which means that the first frame of the shot depicts either the actor performing a significant action or a camera movement. It could also be a combination of both. In that case, we need additional hierarchical groups to describe the action along with the static compositional elements as seen in figure 3.7. This special type of 'action' composition can be due to two actions, camera, actor or a combination of both. If it is a camera movement, we define it under *CameraWith*. In this camera movement, the camera moves along with the subject and does not change the composition. *Pan*, *dolly* and *crane* are some types of camera movements under CameraWith. This type of camera action is also seen in the developments and will be described further under that context. The position of CameraWith is before the composition as we first describe the camera movement that we see and then specify the subject we see with the camera movement, for example, in figure 3.24. The actors can also do the action in relation to the initial composition, and this is defined under *WhileEvent*. In this scenario, we cut



(a) Basis for shot size

(b) Screen coordinates

Figure 3.6: Shot size is a function of the distance between the camera and actors,as well as the camera focal length,as seen in (a). (b) shows the horizontal placement of actors in a composition is expressed in screen coordinates.

into a shot in which the subject is already performing an action, as seen in figure 3.23. WhileEvent is made of the word 'while' and an event. An event is the description of a subject performing an action.

3.1.2. Developments

The third and final part of a shot in our hierarchy is development. There can be one or a series of actions in a shot after the initial composition. It is of two types, *continuation* and *recomposition*. Continuation is when an action, either camera or actor, takes place, but there is no change in the composition from the one before. It can be from an actor *action* such as speaking or looking that is important to the screenplay to mention, but they don't necessarily cause a change in composition. This is categorised under events (figure 3.11). The continuation can also be due to a *follow event* in which the subject starts to perform an action in the *cue*, and the camera moves with the subject and tracks their movements so as to maintain the composition (figure 3.12). Follow events are made of a cue, camera movement called CameraWith and a subject. Generally speaking, a cue is a signal for the actor to start performing an action. We extend this further to include a subject and the action they perform; therefore, a cue

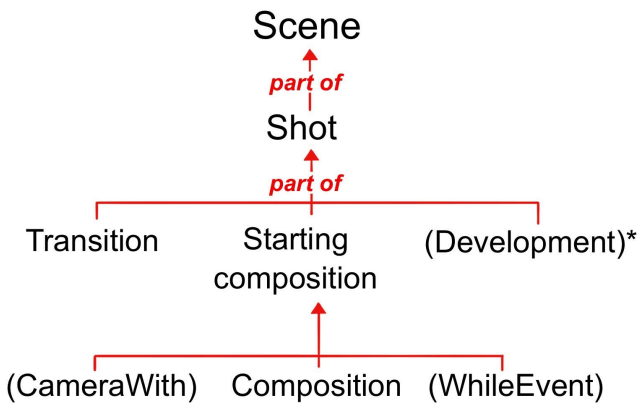


Figure 3.7: Partonomy of starting composition

in PSL describes the actor action. After the cue, we describe the camera action. As it is a continuation, there is no change in composition; the camera movement follows the actor movement closely and hence is called CameraWith. After CameraWith, we specify the subject which the camera is following. Both events and follow events are types of continuation.

The second type of development is recompositions. In these, there is, as the name suggests, a change in composition. This change can either be due to an action performed by the subject or by the camera, or both. Recompositions are made of a cue; camera movement called *CameraTo* and a composition. The cue is the same as discussed before. It is the description of the actor's action. The camera action here is CameraTo, in which the camera can either *hold*, *zoom*, *pan*, *dolly* or *crane*. In the case of hold, the camera does not move, and the change in composition is only due to the actor movement (figure 3.22). After the camera action, instead of only specifying the subject, we describe the new composition in detail as it has changed. The compositional hierarchy is the same as the one we described earlier. There is no limit to the number of developments that can be there in a shot. After the initial composition, there can be any number of developments of either type. To the best of our knowledge, the prose storyboard language is the first description framework that correctly describes developing shots of arbitrary length and complexity shots. Therefore, in 'single shot' movies such as Alfred Hitchcock's Rope, the whole movie can be described in one PSL sentence with multiple developments.

(a) low angle ECU Girl 34left

(b) FS Cyd 34right Fred front

(c) MS Girl 34backright Ferdinand front

Figure 3.8: Single and two actor composition in *Prose Storyboard Language*. Single actor composition (a) from Brian De Palma's *Dressed to Kill*(1980). Compositions from Vincente Minnelli's 1953 musical, *The Band Wagon* (b) and Jean-Luc Godard's 1965 French New Wave film, *Pierrot le Fou* (c) feature two actors in a frame at different sizes.

(a) ECU Scissors MCU Marianne front

(b) CU Sanchez hands front as hands hold Bomb

Figure 3.9: Composition with inanimate objects in *Prose Storyboard Language*. (a) and (b) show the inclusion of inanimate objects in the composition. Both the objects in the compositions, a scissors from *Pierrot le Fou* and the bomb from Orson Welles's *Touch of Evil*, play important roles in the movies and justify their inclusion in the composition.

(a) MLS Father 34right screen left ELS Mother 34left screen right (b) MCU Eve 34left MS Thornhill 34right Vandam 34left Kane 34left screen center MS Thatcher Leonard 34backleft

Figure 3.10: Complex composition with multiple actors in *Prose Storyboard Language*. The frames (a) from Orson Welles's 1941 *Citizen Kane* and from (b) Alfred Hitchcock's 1959 *North by Northwest* show multiple actor compositions at different distances from the camera. They are described from left to right with their sizes indicating their depth in the composition.

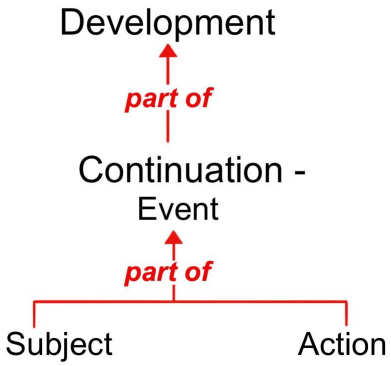


Figure 3.11: Partonomy of a Development type - Continuation with an Event

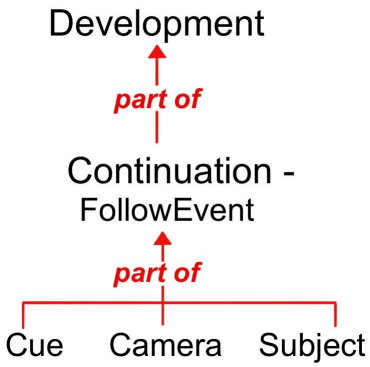


Figure 3.12: Partonomy of a Development type - Continuation with a Follow event

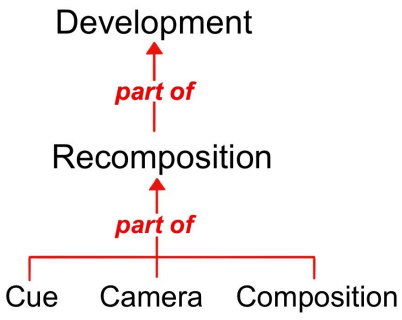


Figure 3.13: Partonomy of a Development type - Recomposition

An essential feature of the language is that all shots are self-contained entities. From the transition to the final composition, each prose storyboard sentence is independent of the previous or the next shot. Shot description can always be written and read without requiring knowledge from the previous or next shot in a movie (figure 3.27).

Based on the taxonomy of shots proposed by Thomson and Bowen [111], there are three main categories of shots :

- A simple shot is taken with a camera that does not move or turn. Any change in the composition is from the movements of the actors in relation to the camera.
- A complex shot is taken with a camera with movements around a fixed point such as pan, tilt and zoom. We introduce camera actions pan and zoom to describe such movements. Thus the camera can pan left and right, up and down (as in a tilt) and zoom in and out.
- A developing shot is taken with a moving camera. We introduce two camera actions (dolly and crane) to describe these shots. Pan and zoom are allowed during dolly and crane movements, thereby creating interesting visual effects.

Our prose storyboard language can describe all these types of shots precisely, as seen in our experiments of annotating scenes from movies.

3.2. Grammar of the Prose Storyboard Language

3.2.1. Syntax and parsing

Grammar is the structure of a language. It is defined as a finite set of rules that are necessary to build a syntactically correct sentence. It is made of two significant elements, non-terminals and terminals. Non-terminals or Auxiliary symbols or syntactic variables are higher-order characters in grammar involved in building a sentence but are not direct components of the final sentence. They are placeholders for the terminals and help us define the rules of the language. Terminals are a string of characters that are the direct components of the sentence. Terminals are generated

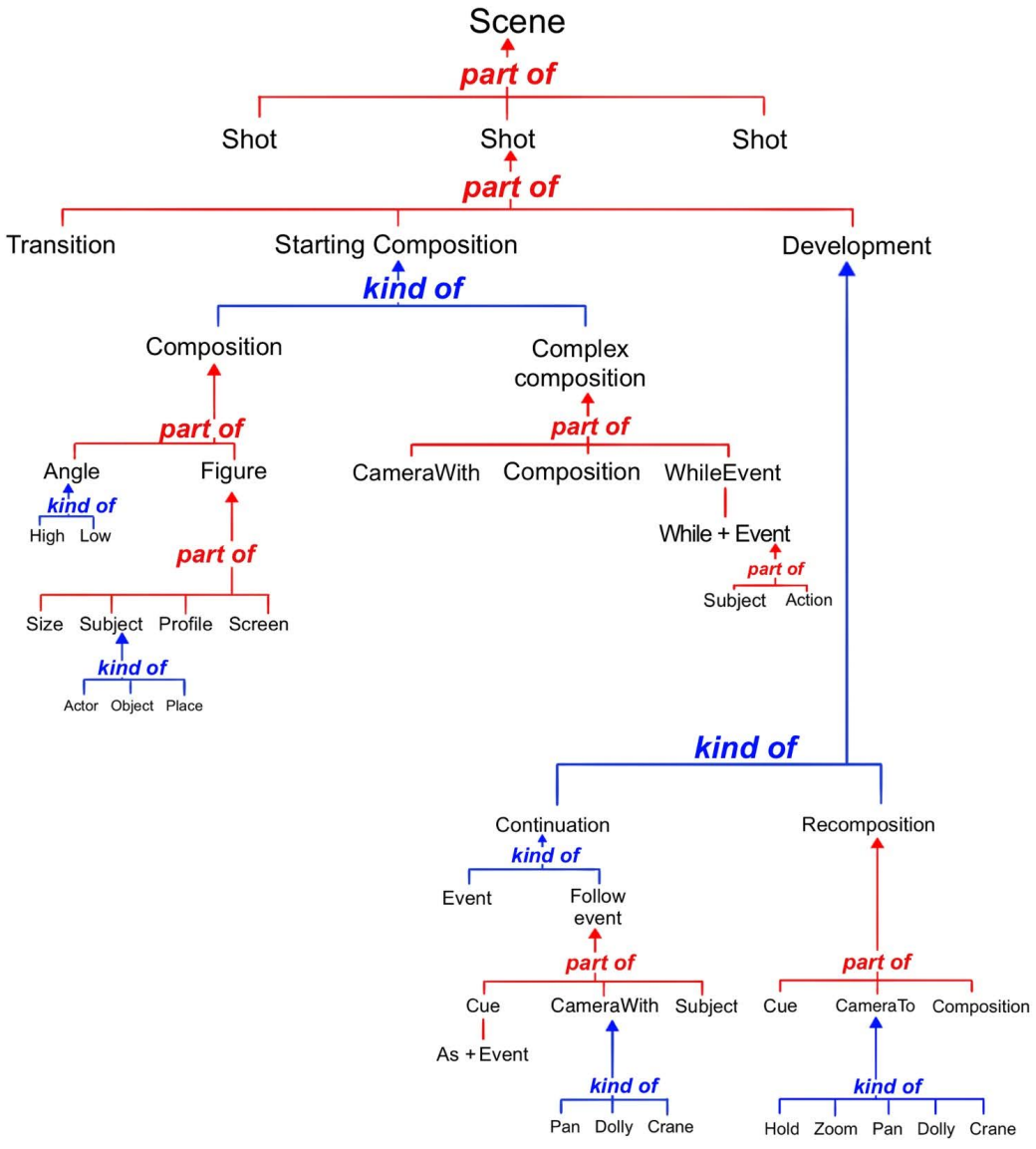


Figure 3.14: Hierarchy of Prose Storyboard Language

PSL term	Definition
Scene	Part of a movie where the action takes place either in a single location or in continuous time or both
Shot	In the final movie, it is a continuous series of frames that are recorded with a camera
Transition	Describes the way pf progression on one shot to another
Composition	Complete description of all the subjects seen in any given frame in relation to the camera
Angle	Specifies the position of the camera in relation to all the subjects in the composition.
Figure	Individual description of each subject in terms of size, profile and screen
Size	Describes the relative size of the subject seen in the frame. It ranges from extreme close up to extreme long shot and is a function of the distance at which the camera is placed from the subject.
Subject	The actor, object or place that is being described in the composition
Profile	The part or side of the subject that is seen by the camera
Screen	Specifies the placement of the subject on screen.
Cue	It is the signal for an event to take place. It includes all the actor actions in PSL
Event	Describes the actor or the object and the action that is performed by them
Development	Series of actions, either camera or actor, that take place after the initial composition.
Continuation	Type of development in which there is action from the actors, camera or both but this action does not change the previous composition.
Recomposition	Type of development in which actions from the actors, camera or both result in a change in composition which is then described in detail.
WhileEvent	Part of initial composition where we directly transition into action. Here, the initial composition is describes while an event is taking place.
FollowEvent	Type of continuation in which the subjects perform a cue action and the camera moves with this cue. The camera follows the event.
CameraWith	Camera action that maintains the composition without any changes.
CameraTo	Camera action that results in a new composition. This includes 'hold' as an action as the camera 'holds' position while the actor action changes the composition.

Table 3.1: Definition of terms

Camera action	Definition
Hold	There is no camera movement
Zoom	The camera does not move but the subject appears larger as a characteristic of the optics of the lens
Pan	The camera is fixed at one point and swivels either horizontally or vertically around this point
Dolly	The camera is sent on a pre-determined track set on an XY plane and moves freely along this track
Crane	The camera is fixed to a crane handle and moves freely in the 3D space

Table 3.2: Definition of terms for camera actions

based on the grammatical rules which dictate where each terminal should be placed in relation to the other. In PSL the non-terminals include the higher order of hierarchy such as a scene, shot, parts of the shot, composition and its parts and development and its parts. The terminals include generic and specific terms. Generic terminals are used to describe the main categories of screen events, including camera actions (pan, dolly, cut, dissolve, etc.) and actor actions (enter, exit, cross, move, speak, react, etc.). Specific terminals are the names of characters, places and objects that compose the image and play a part in the story.

The syntax is the study of grammar rules and the structure of the sentence. It gives us all the information required to build a correct sentence as it outlines the rules of placement of terminals in a sentence. The prose storyboard language is a context-free language. A grammar is said to be context-free if it is just a combination of terminals and non-terminals, and the position of the context of the non-terminal will not affect how that non-terminal can be expanded. In formal grammar, we have a list of rules that define the production of a sentence. These rules show how the components on the right replace the syntactic categories on the left-hand side of the grammar. The complete grammar for the language is illustrated with the AND/OR graph in figure. 3.2 and described in the (Parsing Expression Grammar) PEG notation in figure.3.28. A sentence is a series of terminals or words built systematically by applying the grammatical rules beginning at the highest hierarchy level. So, in our case, a scene is a series of shots; a shot is a combination of transition, initial composition and possible developments. In the simplest case, a shot is a combination of transition and composition. The PSL sentence can be built part by part, for example, “cut to FS Actor

front". It is a complete, syntactically correct PSL sentence. It tells us that we cut into a frame where we see an actor in a full shot from the front.

Parsing of PSL sentences is done in Python using the Parsimonious toolkit [1]. The parser is based on parsing expression grammars (PEGs) in which lexing and parsing are done simultaneously. In language processing, lexing or tokenisation is when the algorithm scans the input and produces corresponding tokens. They divide the input into labelled parts for the parser to analyse. For example, the sentence "cut to FS Actor front" is lexed and produces five tokens, one word per token. On the other hand, parsing is the analysis of the tokens to identify the expression in them and fit that into the syntax tree. When "cut to FS Actor front" is parsed or broken down to determine its syntactic components, we see that it is made of two parts, transition -> cut to and composition -> FS Actor front.

The key feature of PEGs is that it uses a prioritised choice operator "/" rather than an unordered operator "|". This means the order in which the choices are written is important. For example, in a rule 'A = a / b / c ', the parser first checks if the input matches 'a'. It only moves to the next choice if this fails. This prioritisation removes ambiguity and ensures there is only one output parse tree for a given input.

3.2.2. Semantics

On a purely syntactical level, correctly organising the terminals would make a grammatically correct sentence. But this does not ensure that the syntactically correct sentence is meaningful. This is checked by semantics which is the study of the meaning of sentences. To check the semantics of a sentence, we must first ensure that the syntax is correct. A sentence cannot be technically meaningful in language processing if it is grammatically wrong. After the sentence correctly passes syntax checking, we proceed to analyse if it makes semantic sense.

The semantics of the prose storyboard language is best described in terms of a Timed Petri Net (TPN). Petri nets or place/transition nets are graphical, a mathematical tool for formalising distributed systems. The system, in our case, is the runtime of a movie

[1] https://github.com/erikrose/parsimonious

with scenes that contain shots of varying complexity. The semantic representation should describe all the visual elements that take place on the screen. Whether it is a simple shot with no camera movement or a developing shot with an elaborate choreography of actor and camera movements taking place simultaneously, such as the opening shot in Orson Welles' "Touch of evil". The goal of correct syntactic and semantic representation is that a PSL sentence that passes these checks contains all the information needed to create a scene. Petri nets allow us to model PSL sentences and check for semantic meaning graphically. By adding time to Petri nets, we can accurately model the timeline of the shots.

3.2.3. Movie Petrinets

To model PSL sentences as Petri nets, we translate the descriptive elements such as compositions and type of actions and camera movements into *places* and instantaneous actions (such as cuts and the start and end of other events) into *transitions*. The transitions have the time stamps attached to them, and they are fired if all the state requirements leading to the transition are met, and the runtime of the shot reaches the timestamp specified. It is essential to distinguish the transition in a Petri net from that of PSL. Transitions in PSL describe the way we enter a shot. In comparison, transitions in Petri net are actions that, when executed, allow the system to proceed further. Petri net transitions function as checkpoints that indicate the global state of the system based on their firing. It means that we can tell where we are in the video playthrough just by looking at the transitions that have already been fired.

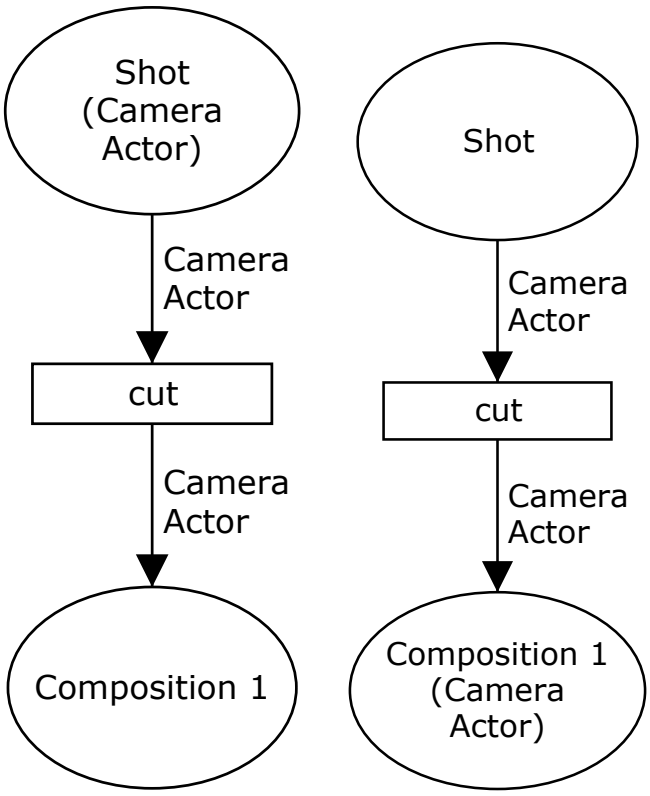
There are three types of Petri nets transitions, first is the transitions from PSL. There are the actions in the net that start the shot. For the sake of simplicity, we only consider the PSL transition of cut while modelling the Petri nets. Cuts are only instantaneous transitions in PSL. The other transitions, fade and dissolve, take time to complete and are beyond this thesis's scope and will be addressed in future work. The second type of Petri net transition are the start of actions, either camera or actor action; these transitions signal the start of these actions. Finally, the end of action Petri net transition signals the end of a camera or actor action. There are two types of tokens, Actor and Camera tokens. These tokens are passed through the net when a transition is fired. An easy example of this is seen in figure 3.15(a) and 3.15(b). When both the

Actor and the Camera tokens are present in one place, we view that as a composition in the video.

In the beginning, when the first cut transition is fired, both the Camera and Actor tokens are passed into the first compositional place. In exceptional compositions where the transition takes us directly into action, we create new places for WhileEvent or CameraWith. When the cut transition is fired, the Camera and Actor tokens are duplicated, and one set of Camera and Actor tokens enter the compositional place. The duplicate Actor token enters the WhileEvent place. This place will have all the frames in which the actor is performing the action described in WhileEvent. The duplicate Camera token enters the CameraWith place, which has all the frames in which the camera is moving. This is seen in figure 3.16. When there is a change in composition, the tokens move further into development. After the initial composition, the development in the shot can either be a continuation or recomposition. In the case of continuation, there is either a camera or actor action, or both, but this does not change the composition. In such a case, after the initial composition, there is a Start transition that sends the tokens into CameraWith and Cue Action places. The camera action tracks the actor action, and at the end, the End transition takes the Camera and Actor tokens back into the same compositional place as seen in figure 3.17. In recompositions, as seen in figure 3.19, there is a change in composition. When the actor starts the action mentioned in the cue, the Start transition is fired, and the Actor token from the first compositional place enters the Cue Action place. This place contains all the frames in which the actor is performing the cue action. Similarly, the Camera token enters the Camera Action place when the Start transition is fired. Then at the end of the camera and cue actions, the End transition is fired, and both the Actor and Camera tokens pass to the second compositional place.

The Petri nets shown here are created using CPN tools [2], a tool to create and simulate high-level Petri nets.

[2] https://cpntools.org/



(a) Before firing of cut (b) After firing of cut

Figure 3.15: Movement of tokens in a Simple shot

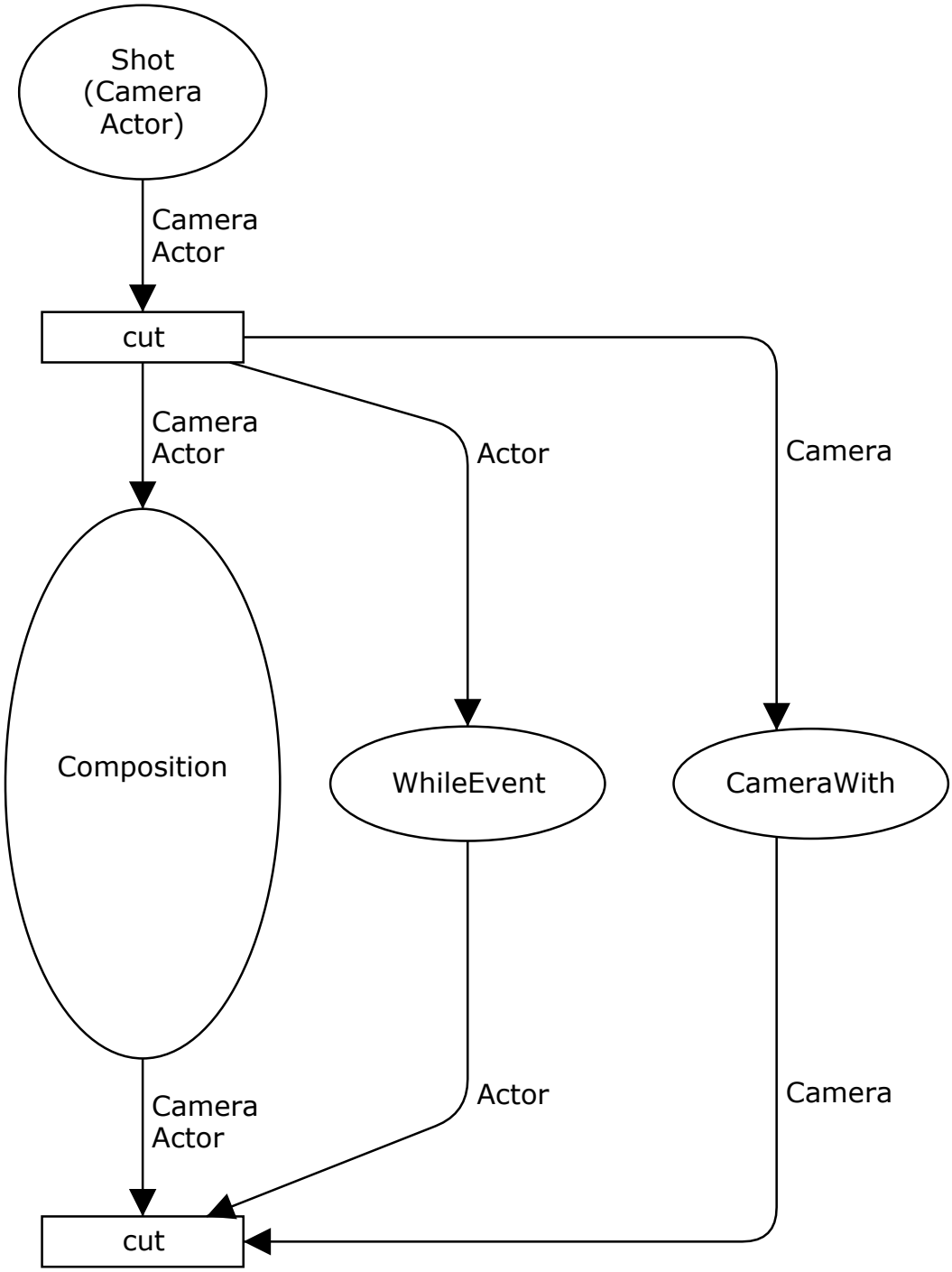


Figure 3.16: Petri Net for initial composition with a camera and actor action

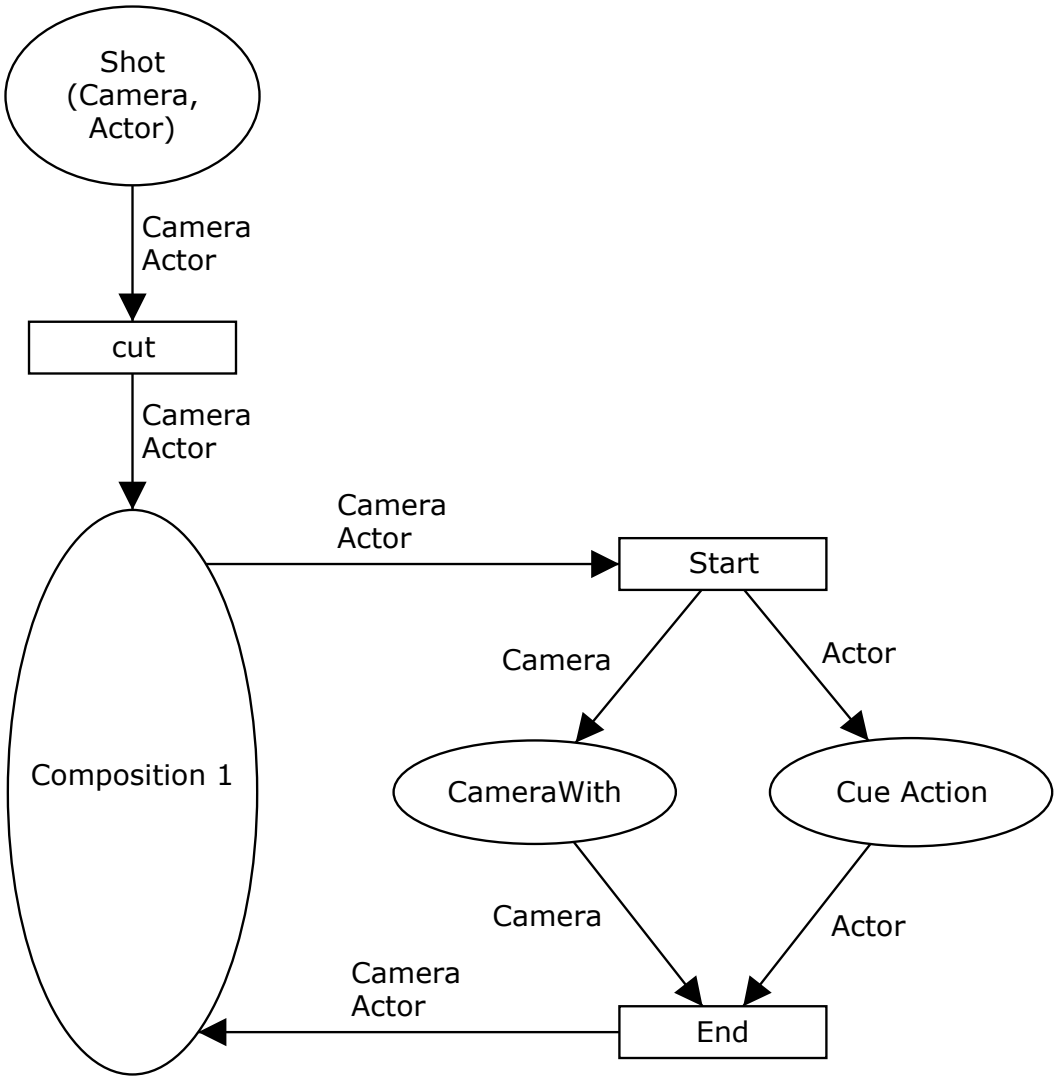


Figure 3.17: Petri Net for composition followed by a continuation

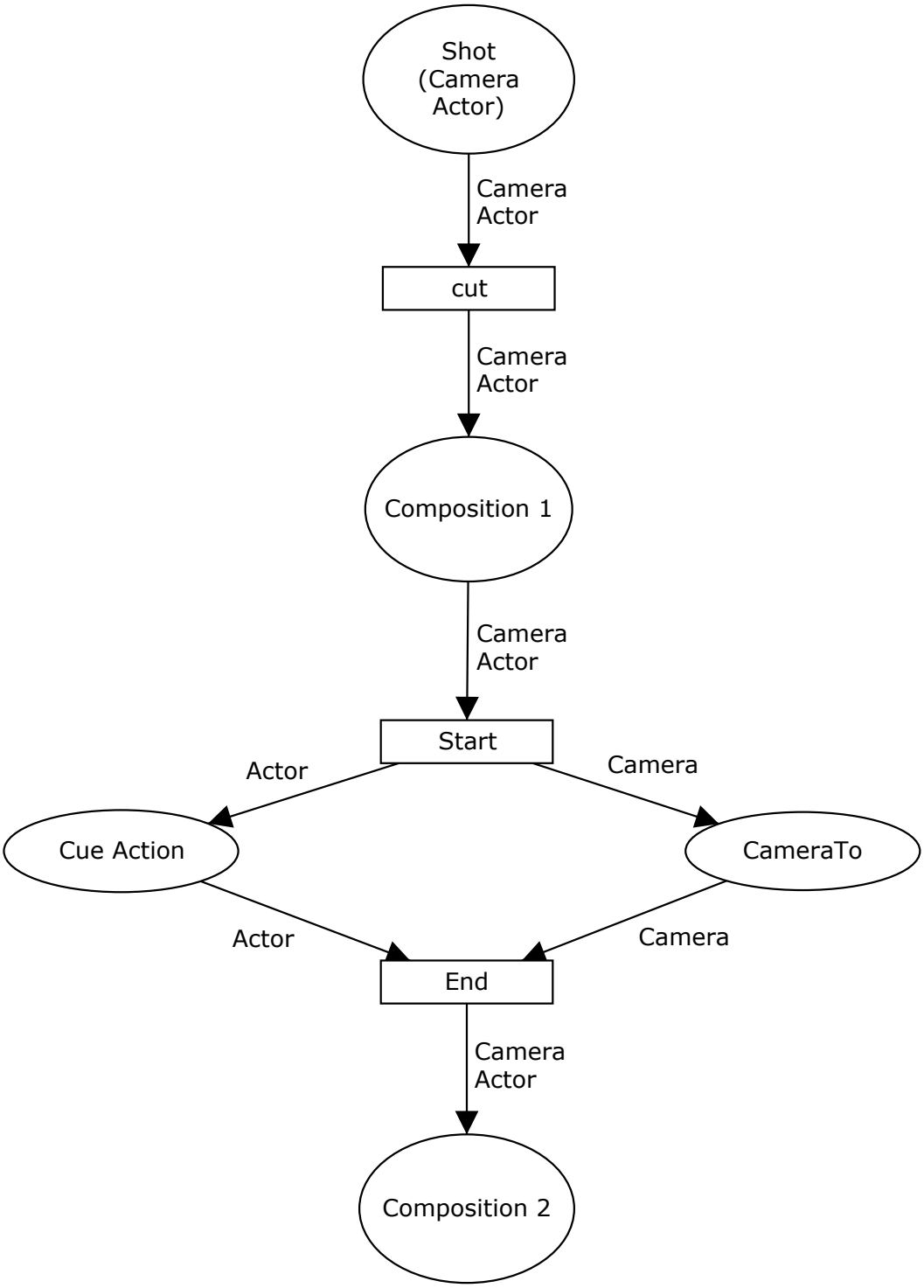


Figure 3.18: Petri Net for recomposition with actor and camera movement

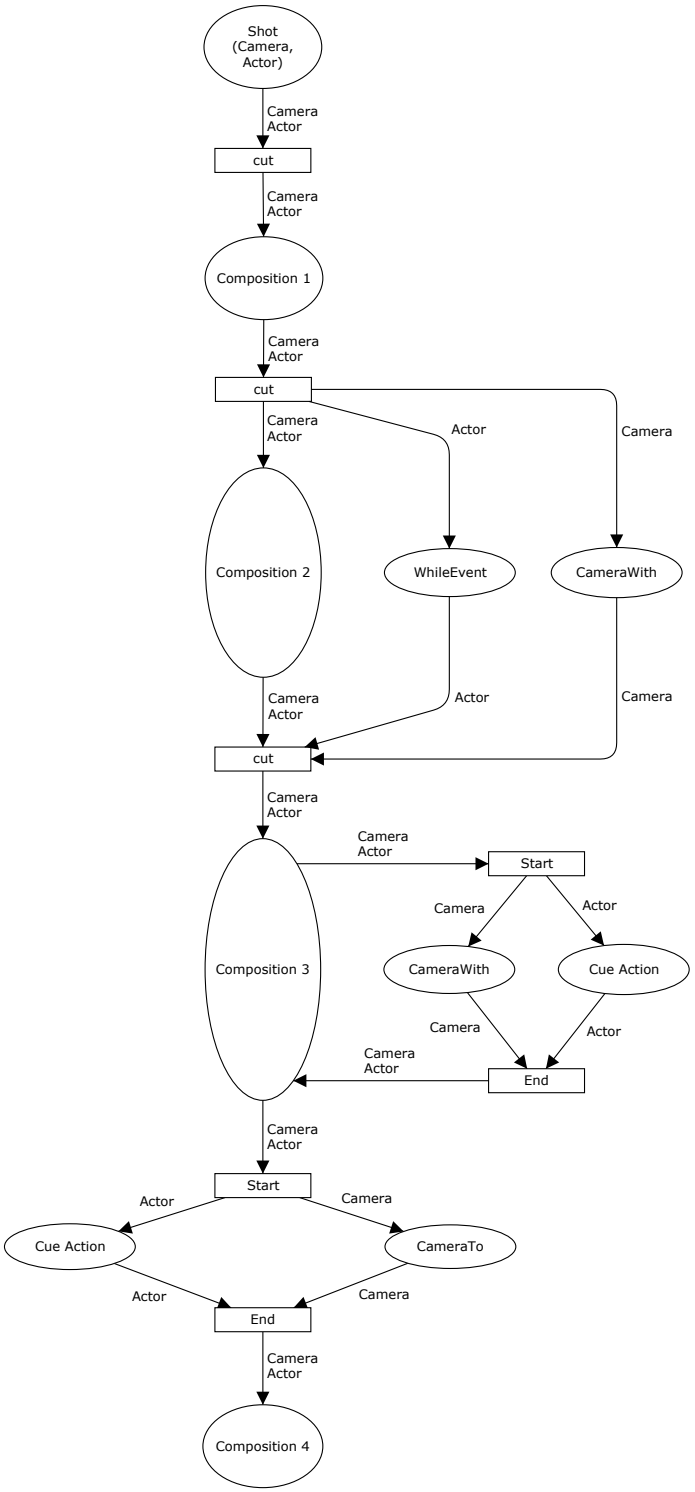


Figure 3.19: Petri Net including all the elements of the Prose Storyboard Language

3.3. Prose Storyboard Language in action

3.3.1. Process of annotation

We start the annotation process by viewing the scenes multiple times. The scene is then divided into its consisting shots, which we describe in Prose Storyboard Language. Each of these sentences is matched to their corresponding keyframe in the shot via timecodes. For example, the first frame of the shot matches the initial composition in PSL. The time code for this keyframe is noted. As the shot progresses, we make a note of the subsequent compositions and their time codes. This list of PSL sentences with time codes for keyframes is written as subtitles in a word processor and saved as SubRip(.srt) files that can be played with the annotated scene. To make the PSL sentences easier to read, they are generally broken down into fragments that start with a 'from' composition and the time codes for the duration. Then the next PSL fragment contains the action that either the camera or actor or both perform that changes this initial composition. After this, we describe the 'to' composition that the previously mentioned action leads to. Their time codes accompany all these fragments. The output of this process of annotation is a time-coded PSL description of the scene in a .srt file.

3.3.2. Annotation results

We annotated scenes extracted from four movies: Back to the Future by Robert Zemekis, Rope and North by Northwest by Alfred Hitchcock, and Touch of Evil by Orson Welles. In each case, we give the original screenplay, the movie subtitled with a complete PSL description of all compositions and developments, and a storyboard with one keyframe per composition or development. Our experimental results are summarized in Table 3.3 and can be found in the accompanying material [3]. With 177 shots and 330 compositions, they constitute an informal validation of the expressivity and generality of the language and an illustration of good practices for precisely annotating movie shots using the language.

[3] https://team.inria.fr/anima/prose-storyboard-language/

The cafe scene in Back to the future (1985) focuses on dialogues between 8 characters. We are making the prose storyboard for all 41 shots in the entire scene available for future reference. Rope (1948), a single shot movie by Alfred Hitchcock, shows a dialogue between 8 characters using elaborate blocking and camera movements rather than cuts. This is a challenging example for annotation, and we show examples from two extended sequences fully annotated with PSL. Results are shown in figure 3.27.

We also annotated the crop duster attack scene from North by Northwest (1959) to highlight the versatility of the language in describing a scene with non-human actors in an outdoor environment. In that scene, the intent of the pilot is personified in the movements of the plane. We annotated all 133 shots in this virtuoso scene with their prose storyboards to illustrate the variety of shots used in this mostly silent scene. Finally, we annotated the long opening shot from Orson Welles's Touch of Evil (1958), which shows a wide variety of camera movements interlaced with meticulously planned choreography for the characters resulting in a rich and dynamic visual composition. Despite the complexity of these scenes, we show that the prose storyboard is fairly simple to read and easy to generate.

3.4. Summary

We present a language for describing the spatial and temporal structure of movies with arbitrarily complex shots. We outline its syntax and propose a semantic model. We showcase its real-world usability by annotating scenes from movies in PSL. As we show in the subsequent chapters, the language can be extended and adapted to specific use cases. We also believe that the proposed language can develop existing approaches in intelligent cinematography and editing towards more expressive strategies and idioms and bridge the gap between real and virtual movie-making.

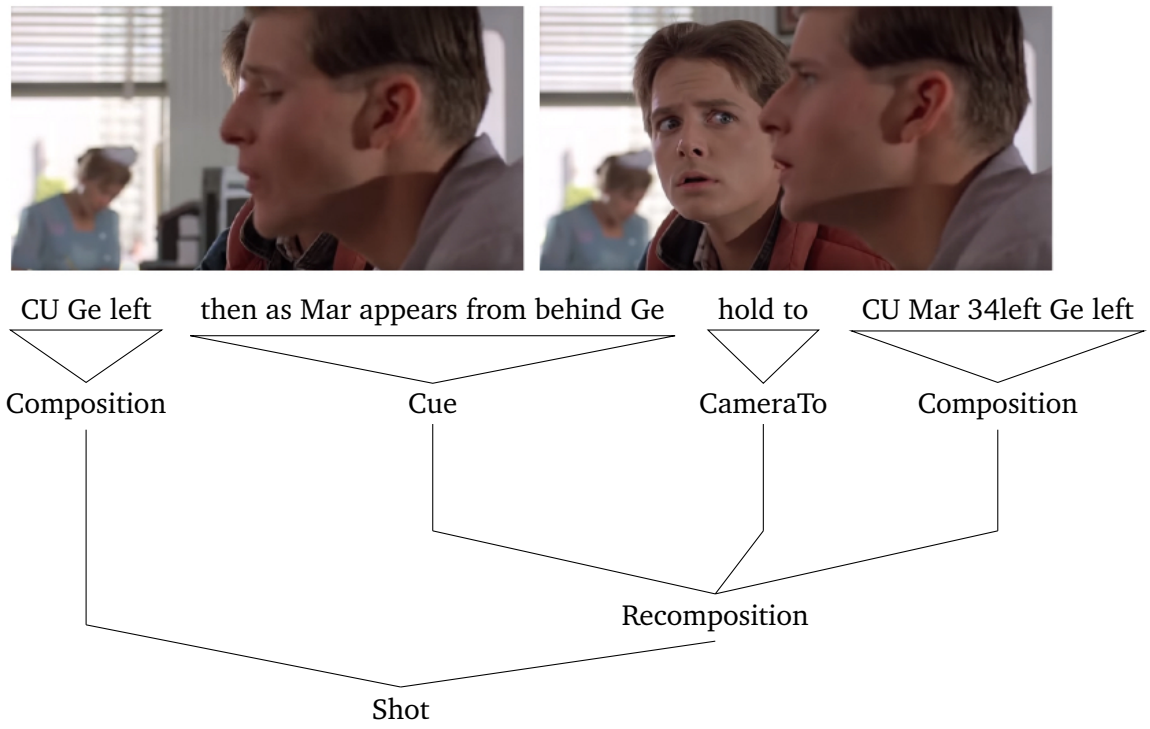


Figure 3.20: Simple shot with actor movement in *Back to the future*

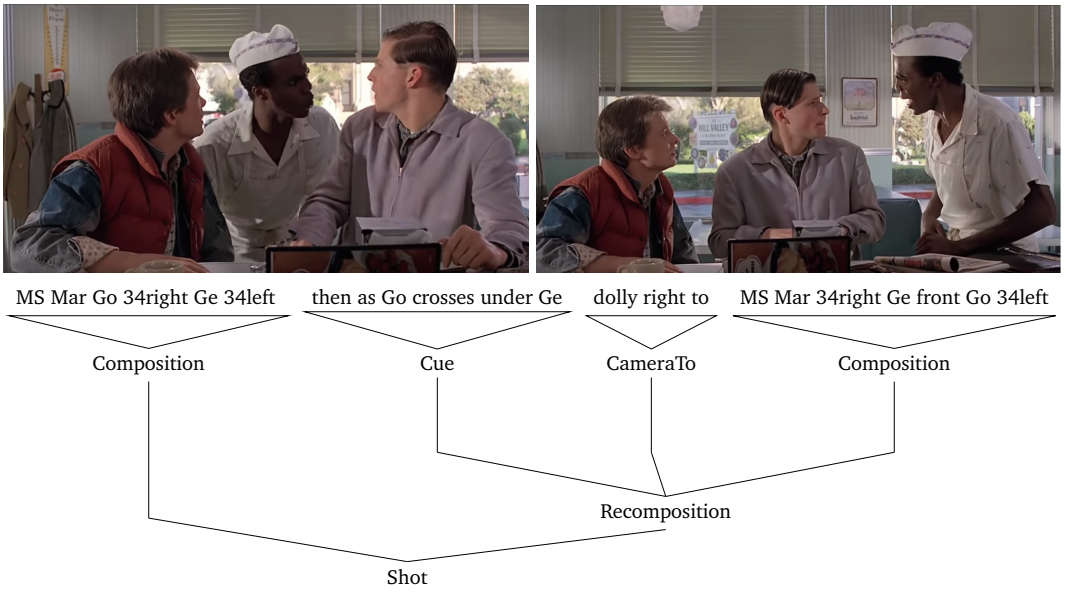


Figure 3.21: Developing shot in *Back to the future*

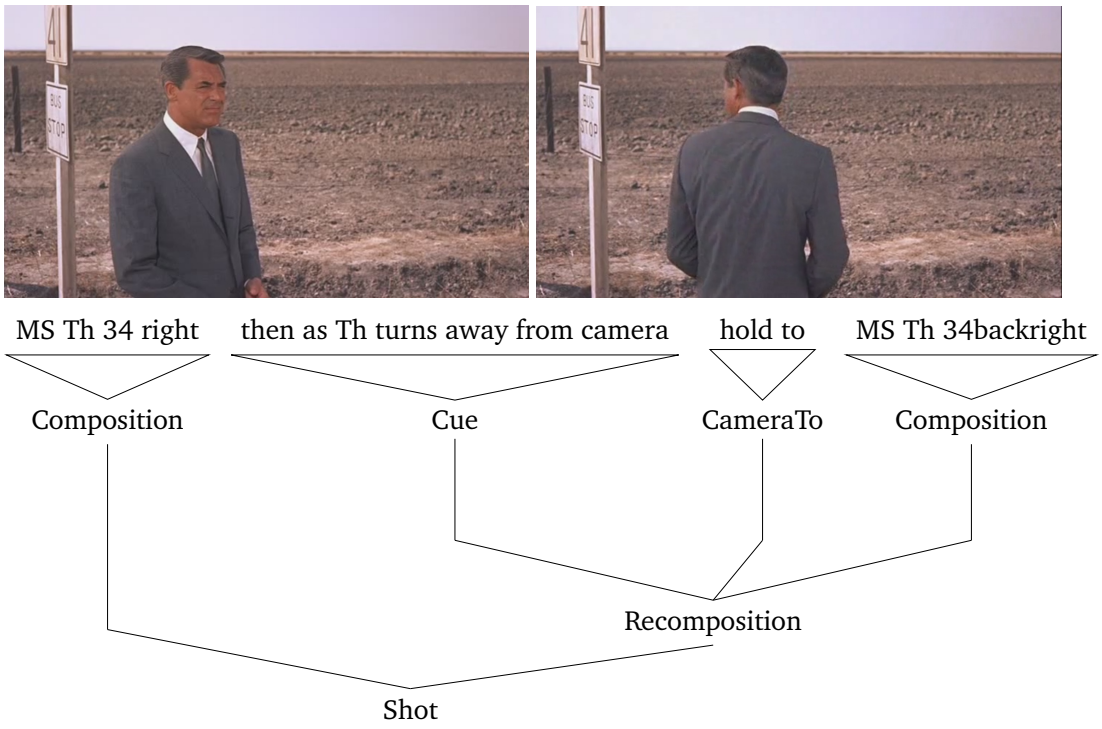


Figure 3.22: Simple shot: with actor movement in *North by Northwest*

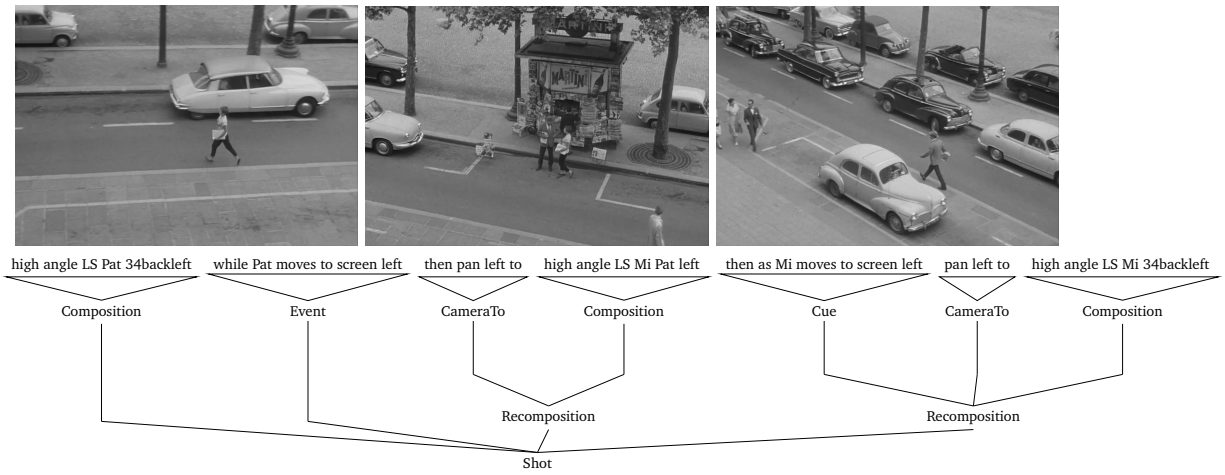


Figure 3.23: Complex shot in *Breathless*

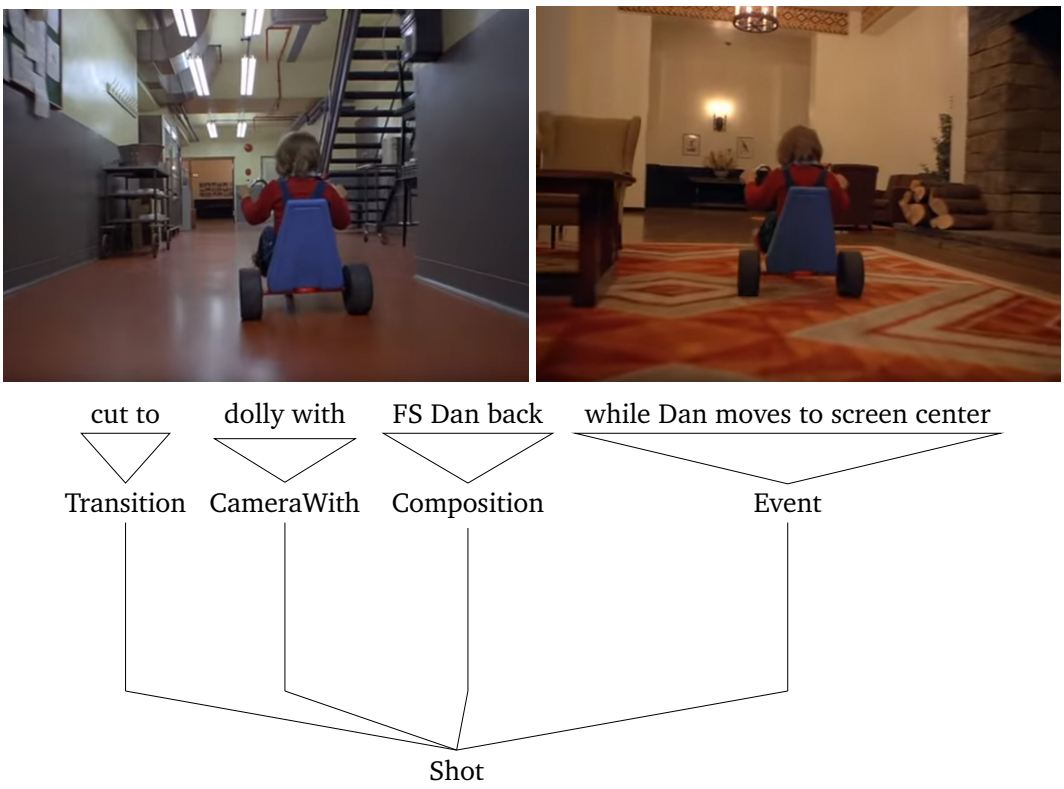


Figure 3.24: Developing shot: Dolly with actor in *The Shining*

Movie	**Shot**	**Composition**	**Development**							
			Continuation			Recomposition				
			Pan	Dolly	Crane	Hold	Zoom	Pan	Dolly	Crane
Back to the Future	41	69	-	-	-	12	1	6	10	-
North by Northwest	133	209	2	7	-	49	1	8	16	-
Touch of Evil	1	40	-	2	1	6	-	1	18	13
Rope	2	12	-	1	-	-	-	3	7	-
Total	177	330	2	10	1	67	2	18	51	13

Table 3.3: Annotation results: For each movie, we give the total number of annotated shots, compositions and developments, together with a count of the main categories of camera movement.

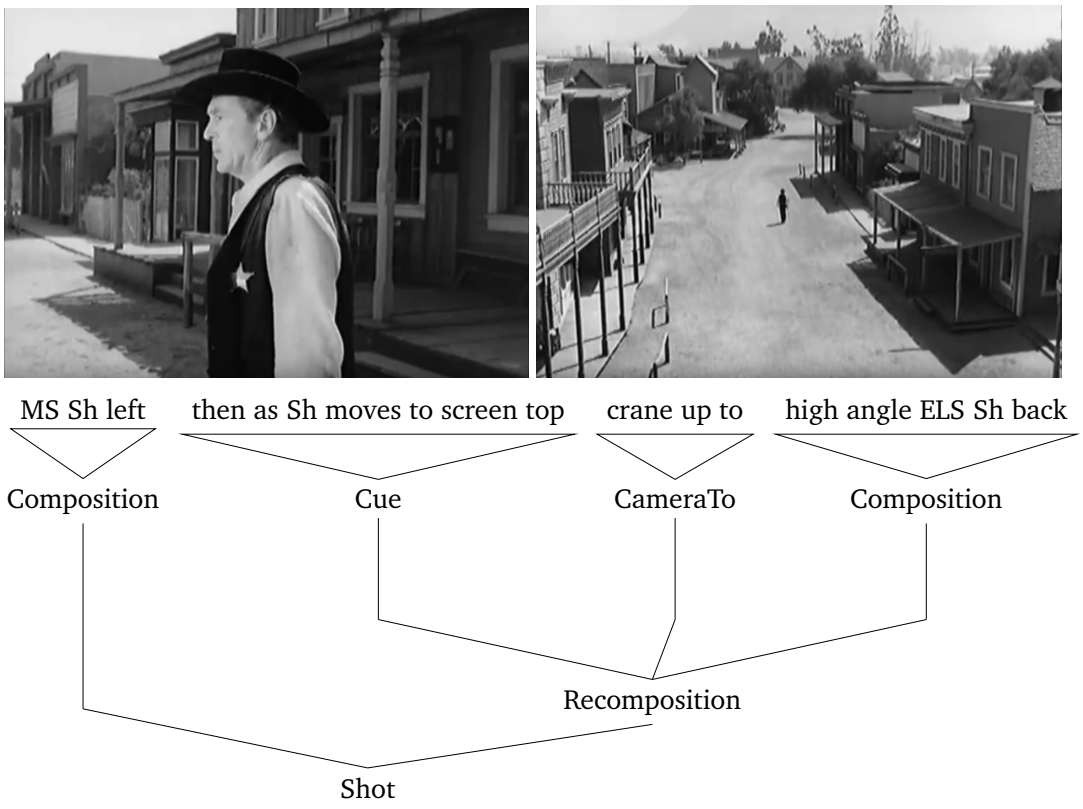


Figure 3.25: Developing shot: Crane up in *High noon*

Shot size	**Definition**
ECU	Extreme Close Up
CU	Close Up
MCU	Medium Close Up
MS	Medium Shot
MLS	Medium Long Shot
FS	Full Shot
LS	Long Shot
ELS	Extreme Long Shot

Table 3.4: Abbreviations of shot sizes

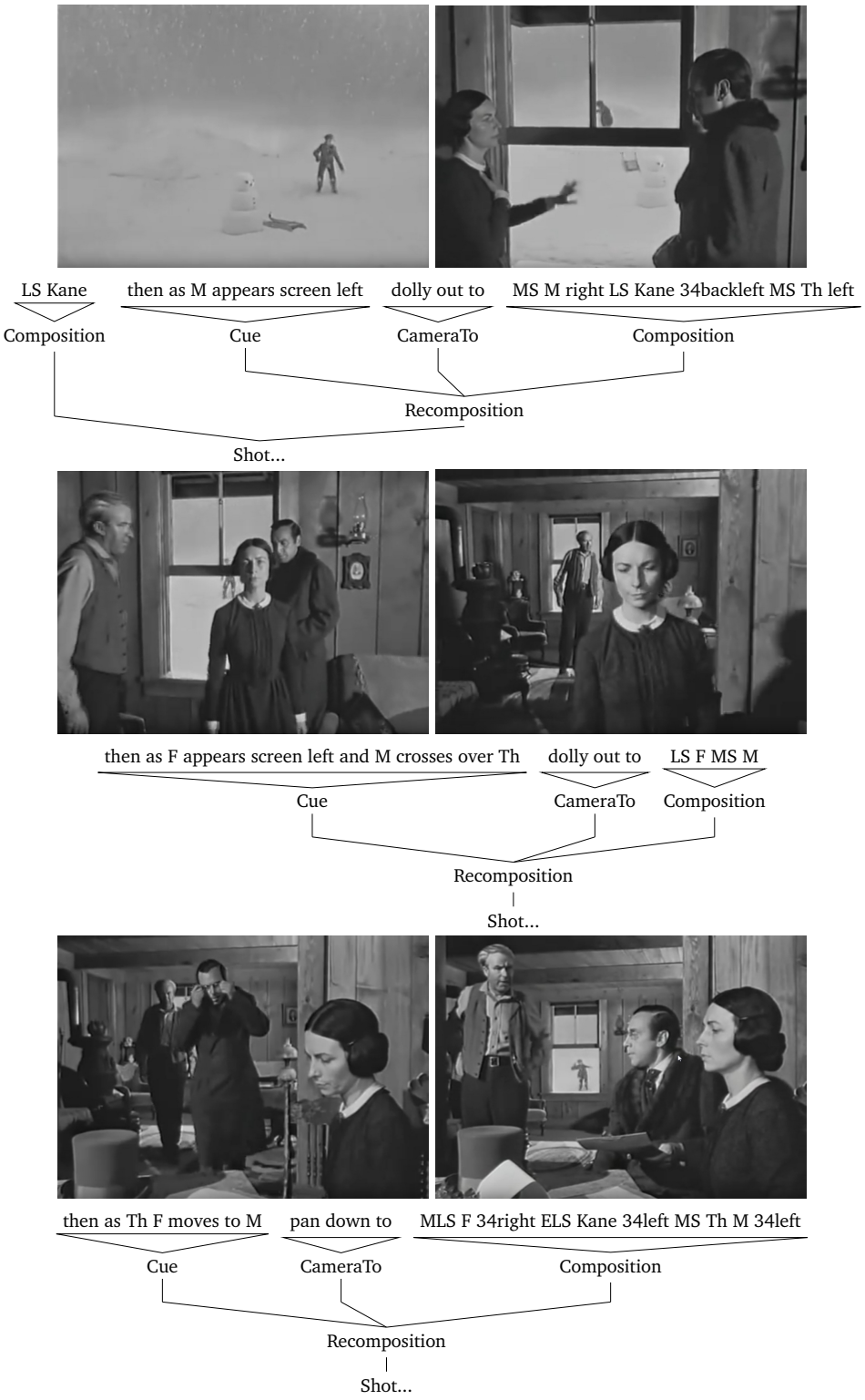


Figure 3.26: Developing shot with multiple actors in *Citizen Kane*

Figure 3.27: Prose storyboard language annotations of two extended sequences from the movie *Rope*.

```
Scene          = Shot*
Shot           = Transition? _ CameraWith? _ Composition? _ WhileEvent?
                  _ (Development?)*
Transition     = ("cut to" / "dissolve to" / "fade in to")
Development    = "then"? _ (Recomposition / Continuation)
Continuation   = Event / FollowEvent
Recomposition  = Cue? _ CameraTo _ Composition
WhileEvent     = "while"_ Event
FollowEvent    = Cue? _ CameraWith _ Agent _ ("and"? _ Agent)*
Cue            = "as" _ Event _ ("and"? _ Event)*
Composition    = (Angle? _ Figure)*
Figure         = Size? _ Subject _ Profile? _ Screen?
Agent          = (Actor / Object) _ (Actor / Object)*
Subject        = Actor / Object / Place
Angle          = "low angle" / "high angle"
Size           = "ECU" / "CU" / "MCU" / "MS" / "MLS" / "FS" / "LS" / "ELS"
Profile        = "left" / "right" / "front" / "back" / "34left" / "34right"
                   / "34backleft" / "34backright"
Screen         = "screen" _ ("top" / "bottom")? _ ("left" / "center"/ "right")?
CameraWith     = Speed? _ (Pan / Dolly / Crane)  _ "with"
CameraTo       = (Hold /(Speed? _ (Pan / Dolly / Crane / Zoom))) _ "to"
Hold           = "hold"
Pan            = "pan"   _ ("left" / "right" / "up" / "down")?
Dolly          = "dolly" _ ("left" / "right" / "in" / "out")?
Crane          = "crane"  _ ("up" / "down") _ ("left"/"right")?
Zoom           = "zoom" _ ("in" / "out")
Speed          = "slow" / "quick"
Enter          = "enters"  _ Screen? _ Place?
Exit           = "exits" _ Screen? _ Place?
Look           = "looks" _ "at"? _ (Subject / Screen)
Move           = "moves" _ "to" _ (Subject / Screen )
Speak          = ("speaks" / ("says" _ String)) _ ("to" _ Subject)?
Use            = "uses" _ Object
Cross          = "crosses" _ ("over" / "under") _ Subject
Touch          = "touches" _ Subject
React          = "reacts to" _ Subject
Turn           = "turns" _ ("left" / "right"
                   / "towards camera" / "away from camera")
Stop           = "stops" _ ("at" / "near") _ Subject
Appear         = "appears" _ (Screen / ("from behind" _ Subject))
Disappear      = "disappears" _ (Screen/ ("behind" _ Subject))
Action         = Enter / Exit / Look / Move / Speak / Use / Cross / Touch
                  / React / Turn / Stop / Appear / Disappear
Event          = Agent _ Action
```

Figure 3.28: Grammar of the prose storyboard language in the Parsing Expression Grammar (PEG) format.

```
Actor       = "Thornhill" / "MBS" / "Plane" / "TD1" / "TD2" / "Farmer" /
              "BC Driver" / "BC Woman" / "BC Man"
Object      = "Bus" / "White car" / "Limo" / "Truck" / "Blue car" /
              "Green bus" /"Bluewhite car" / "Oil truck" / "Pickup"
               / "Brown car"
Place       = "Arid plot 1" / "Arid plot 2" / "Arid plot 3" /
              "Corn field" / "Highway" / "Dirt road"
```

Figure 3.29: Script elements for North by Northwest.

```
Actor       = "Brandon" / "Philip" / "Atwater" / "Janet" /
               "Kentley" / "Kenneth" / "Rupert"  / "Wilson"
Object      = "Glass"
Place       = "Salon" / "Dining room" / "Kitchen"
```

Figure 3.30: Script elements for Rope.

```
Actor       = "Marty" / "George" / "Biff" / "Lou" / "Goldie"
              / "Match" / "Skinhead" / "hands" / "3D"
Object      = "Coffee"/ "Bar" / "Car"
Place       = "Cafe"
```

Figure 3.31: Script elements for Back to the future.

```
Actor       = "Kane" / "Mother" / "Thatcher" / "Father"
Object      = "Papers"/ "Window"
Place       = "Outside"
```

Figure 3.32: Script elements for Citizen Kane.

```
Actor       = "Mike" / "Susan" / "Linnekar" / "Blonde" / "Sanchez" /
               "Immigration official" / "Customs official"
Object      = "Car" / "Bomb" / "Building" /"Checkpost"
Place       =  "Border control" / "Main street" / "Parking lot"
               / "Left side street"
```

Figure 3.33: Script elements for Touch of Evil.

Chapter 4

Anatomic scene generation

As we have established a formal cinematographic language in the previous chapter, we now describe the Text-to-Movie authoring system that uses this language as its input. We selected anatomy as the use case for its pivotal role in medical education and the difficulty of teaching and learning the subject. Expanding the language beyond cinema and into pedagogy also highlights the adaptability of PSL. Our authoring tool takes scripts written in the formalised cinematographic language and converts them into Hierarchical Finite State Machines read by a Unity application to generate an animated video. Narration can be added to these lessons after they are scripted in ASL (figure 4.1). In this chapter, we describe the anatomical extension of the Prose Storyboard Language. Then we explain how we translate the information from these scripts into state machines.

4.1. Anatomy Storyboard Language

As previously stated, our system's input is text, written in a formal language called the Anatomy Storyboard Language (ASL). It is a domain-specific language in which the video produced is registered as a set of unique sentences. Each sentence describes all the visual elements, camera actions and animations seen from the start of the recording till the camera stops. As each sentence can generate a complete shot, it must have all the information necessary to transition into the shot, build the composition,

Figure 4.1: Text-to-movie generation example. From left to right: input ASL script; automatically generated animation; optional narration added by anatomy expert during lesson.

direct camera movements, record all the developments from the initial composition and finally describe the last composition before the camera stops. Figure 4.2 shows the detailed And/Or graph of the grammar for Anatomy Storyboard Language.

The partonomy and taxonomy are similar to Prose Storyboard Language, with a few details specific to anatomy. The video, as in PSL, is made of a series of scenes. A scene is made of a series of shots. A shot has three parts, transition, initial composition and developments. Transition describes how we enter the shot. In anatomy, we primarily use the simple 'cut' transition.

The initial composition is the complete description of the visual elements in the first frame of the shot. It is made of angle and figure. The angle is the position of the camera in the Y-axis in relation to the subject being viewed (figure 4.3(a)). *Figure* term consists of the detailed description of the anatomical part or region seen on the screen. First, we mention the size. It specifies the extent of the subject seen within the camera frame (figure 4.3(b)). Describing the size for anatomical subjects requires the mention of the specific parts in focus. In Prose Storyboard Language, it is taken as default that a close up would mean the focus is on the actor's face unless otherwise mentioned. This assumption works in PSL as the actor's face is generally the most important feature to focus on in a scene. In ASL, the user has to specify the point of interest as one cannot make any assumptions about the importance of one part over the other. For example, a close of Femur could mean different things based on the

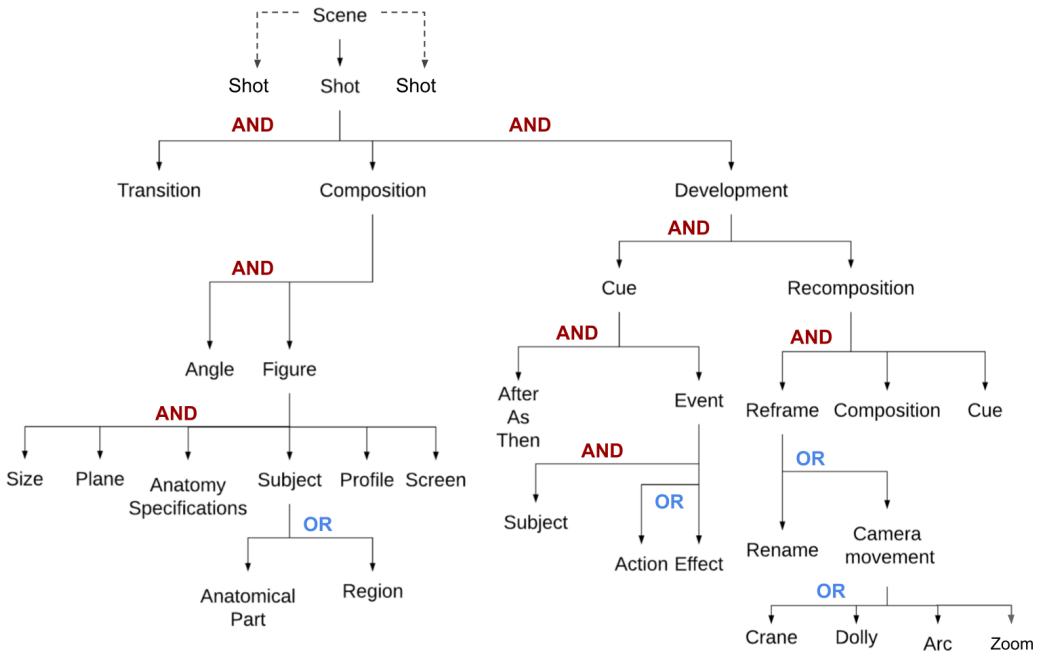
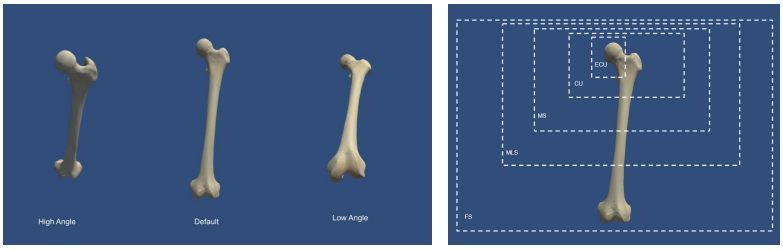


Figure 4.2: And/Or Graph representation of the Anatomy Storyboard Language grammar. ASL scenes are made of shots containing an initial composition and one or more optional developments.



(a) Angles (b) Anatomy sizes

Figure 4.3: Angles and sizes in Anatomy Storyboard Language

lesson. When teaching the knee, the author must specify that the close up should be on the distal condylar part of the knee and not the acetabulum that is part of the hip joint.

After describing the size, we describe the *plane* in which we view the subject. Anatomically, there are three major planes. These planes are hypothetical 2D flat surfaces that run through the body. They are *frontal*, *sagittal* and *transverse* planes as seen in figure 4.4. The frontal plane runs from side to side and divides the human body into ventral or anterior and dorsal or posterior parts. The sagittal plane runs from the front to the back and divides the body into left and right parts, and the transverse plane is horizontal and divides the body into upper and lower parts. In ASL, the camera is placed perpendicular to the planes mentioned to see the anatomical part in that plane. For example, if we view the ventral or front part of the Femur or the thigh bone in frontal view, the camera is perpendicular to the frontal plane that runs from side to side. Therefore we see the frontal view of the Femur. In practice, most of the anatomical parts are viewed along the vertical axis. So, either sagittal or frontal planes. It is not necessary to mention them as that information can be extracted from the *profile* description, as we shall see. But if the desired composition is in the horizontal axis(transverse), it must be mentioned in composition.

The next part of the composition is *Anatomical specification*. The human body is bilaterally symmetrical, which means that the general shape and the anatomy of the limbs are mirrored in the sagittal plane. It is also called left/right symmetry. It means that the left and right sides are symmetrical but mirrored, so it is crucial to mention which side we see in the composition. The first part of the anatomical specification

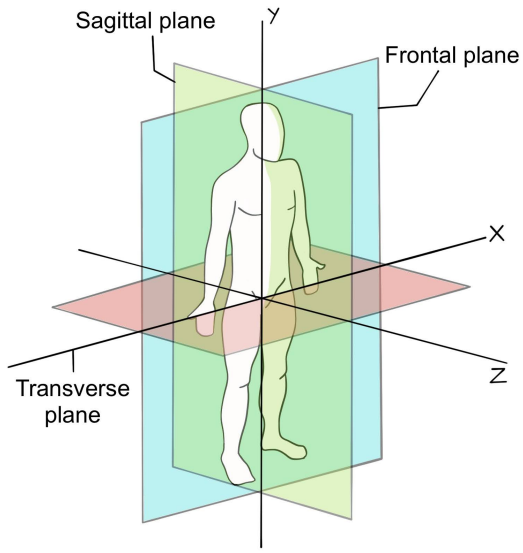
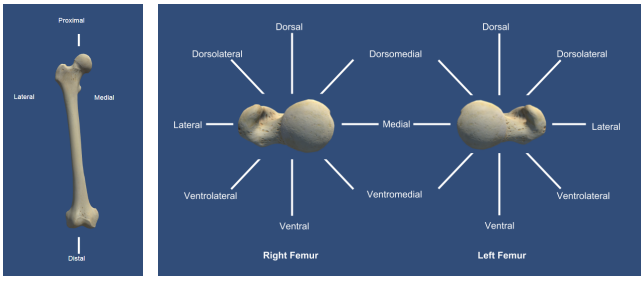


Figure 4.4: Anatomical planes

is to note the side to which the subject belongs. The second part is necessary if the subject is viewed in the transverse or horizontal plane. In this case, we specify if we see the subject from the proximal end (close to the centre of the body) or the distal (further away from the centre) as seen in figure 4.5(a).

The subject, which are either anatomical parts or a region, are described after the anatomical specifications. Then we describe the profile, which is the orientation of the part in relation to the camera. It is the side of the subject that is viewed by the camera. The anatomical profile is different from PSL due to bilateral symmetry and is shown in figure 4.5(b). The last part of the composition is the screen that describes the subject's position in terms of screen coordinates. For now, we concentrate on teaching one anatomical region at a time, and this region will automatically be centred on the screen.

Developments in ASL have a simpler structure than in PSL. They are the changes in the visual elements from the initial composition. As in PSL, there are no limits



(a) Specifications (b) Profiles in transverse plane

Figure 4.5: ASL specifications and profiles.

placed on the number of developments that can take place in a shot. A development in ASL is made of two parts, the *cue* and the *recomposition*. A cue has an event that describes a subject performing an action or an effect concerning the subject. Examples of actions are the movements of the joints, such as flexion/extension. This cue can be written as "... then as the Knee flexes completely ...". Effects are the appearance and disappearance of parts, for example, "... then as Tibia appears ... ".

The camera action is similar to that of PSL. ASL has an additional camera action called arc, a dolly on a circular track around the anatomical part. This movement is the most used camera action as we view the subject from different profiles. After there is a cue or camera action, there is a recomposition.

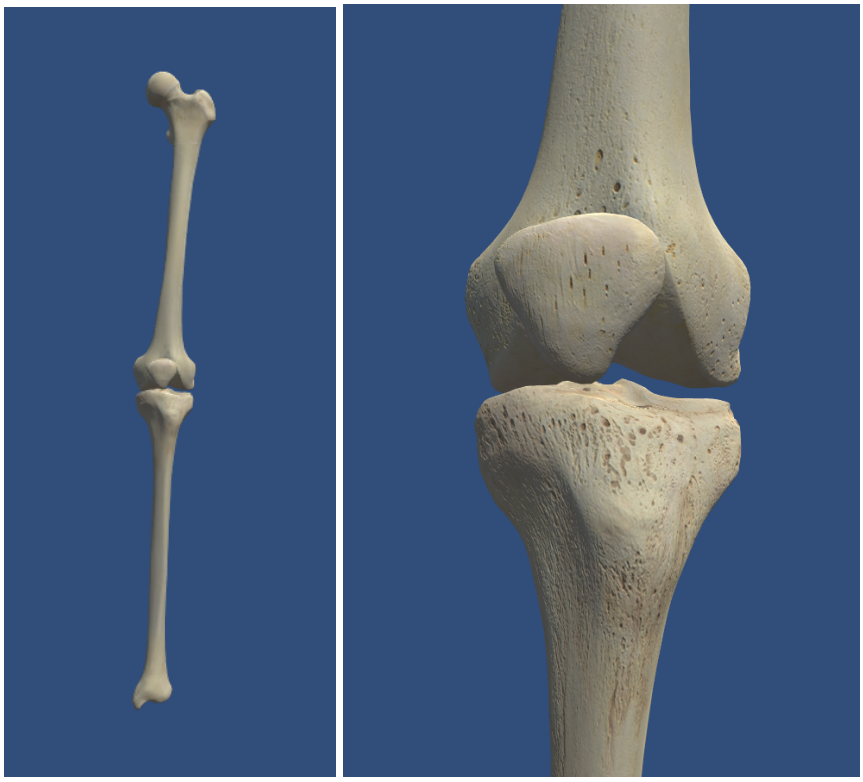
4.2. Building the 3D anatomical scenes: ASL to Hierarchical Finite State Machines

ASL scripts are parsed using the same Parsimonious Python library as PSL. Figure 4.7 outlines the ASL grammar. The semantics for ASL are modelled as state machines, which means that the different elements written in ASL are organised into states and transitions of a Hierarchical Finite State Machine (HFSM). These state machines are written in an Extensible Markup Language (XML) file.

Anatomy videos are less complicated than regular movies as all the parts are connected, and any action performed is applied to all parts simultaneously. This eliminates the need to describe every actor separately, which means all parts can be described as one entity. This and further simplification of the camera movement usually seen in anatomical videos means that the video is more or less linear and does not require the concurrency of the Petri net models. For this reason, we chose the semantic model to be Hierarchical Finite State Machines. They have an added advantage that the Unity engine we use to visualise the lesson runs on finite state machines.

State machines are abstract behavioural models that represent the complete system in terms of states and transitions. States are parts of the system which take an input and perform the action written for that state to give an output. Transitions connect the states and have rules that the system needs to fulfil to progress from one state to another. State machines are directed graphs with the nodes representing states and the arrows representing the transitions. Since the system has a finite number of states in most cases, this modelling system is also referred to as finite-state machines (FSM) or finite-state automata. Hierarchical finite state machines are FSMs whose individual states themselves can be expanded to form other state machines. They are nested states or superstates of ordinary FSMs. This type of hierarchy within states allows us to organise tasks that the state/ transitions perform at different levels. At the higher level, you have more general tasks being carried out, which in our case is progressing through a shot at the composition and development level. Then at a lower level, such as within a hierarchical state, we can expand the development into cue actions and recompositions. At the lower state level, the system performs the general task of going through the shot and the local task of progressing through development.

The state machines are built part by part using descriptive tags. A complete ASL sentence or a shot, is contained within a *Scenario* tag which contains a list of *States* and *Transitions*. The transitions mentioned with HFSMs are different from the transitions mentioned in PSL or ASL. Here, transitions refer to the unilateral paths between states that direct progression in the system. Each *State* is given a unique *Name* and contains an *Anatomy* list of parts that present in the current composition. These are the ASL subjects. The state in HFSMs also contains other information about the composition, such as the angle, plane and profile. They are all stored under the *camera* tag with



(a) FS Left Femur Tibia Patella ventral

(b) CU Left Femur Tibia Patella ventral

Figure 4.6: Sizes in HFSM

the state. The camera also has a *Lookat* tag that lists the objects the camera should look at. This list is stored under a *group* tag. In HFSMs, we use this Lookat tag to store information about the sizes. If the ASL calls for a full shot (FS), the *Lookat* will include all the anatomical parts mentioned in the composition. Here the list of parts in state and the Lookat tag are the same. This way, the parts mentioned are seen with all their edges within the camera frame. In the case of a close up (CU), the *Lookat* list will exclude the large parts to focus on the smaller anatomical parts. The list of parts in the state is more than the list of parts in Lookat. So in the final visualisation, the parts mentioned in the state are seen, but the close up focuses on the parts mentioned in the Lookat list. In that case, bigger anatomical parts are still present, but portions of them are left offscreen. An example is given in figure 4.6 for a full shot of the left femur, tibia and patella and then a close up on the patella.

The second part of the HFSM is transitions. They are described under the *Transition* tag, and in this, we specify the input and the output state for that transition. Transition also contains two addition tags, *delay* and *time*. The delay tag refers to the amount of time the system spends in the input state. The time tag refers to the time it takes for the camera movement to be executed if any camera movement is mentioned.

As the shot develops, there will be changes in the composition. These changes can be due to *actions* or *effects* in *cues*, or *camera movements*, or both. *Actions* in ASL are translated into animations in the HFSM. An additional *Animation* tag is added in the state to trigger animations of anatomical elements (e.g. a knee flexion). In the current state of the application, animations are pre-made and cyclical. There is a library of premade animations, and based on the animation mentioned in the script, the corresponding animation tag gets added to the state. The animations are cyclical, which means each animation starts and ends in the default anatomical position of the human body. Each animation is complete, and the body position does not alter after the animation ends. This is done to avoid glitches. For example, if an animation, such as knee flexion, is mentioned in state A and the animation only performed the flexion action, the knee would then remain in the state of flexion. This means that in the next state, state B, the user must not forget to include an animation of knee extension that allows the 3D model to return to its anatomical position. Suppose the user forgets this or changes it in editing without paying attention to the state before. In that case, it will lead to glitches in the final video, with the knee jumping from a state of flexion to its regular position.

4.3. Summary

This chapter describes the Anatomy Storyboard Language, which forms the basis for our authoring system. It is a use case extension of the Prose Storyboard Language with anatomical specifications. It is parsed similarly to PSL, and the outcome of the parsing is translated into a Hierarchical Finite State Machine. The state machine is then read by the Unity application, which builds the 3D scene and animates it according to the instructions written in the ASL script.

```
Shot            = Transition? _ Composition? _ (Development)*
Development     = Cue? _ Recomposition?
Recomposition   = Reframe? _ Composition? _ Cue?
Reframe         = Rename/CameraMov
Cue             = After / As / Then
After           = "then"? _ "after" _ Time? _ Event?
As              = "then"? _ "as" _ Event _ ("and"? _ Event)*
Then            = "then" _ Event? _ ("and"? _ Event)*
Composition     = Angle? _ Figure _ (Angle? _ Figure)*
Figure          = Size?  _ Plane? _ AnatSpecs _ Subject _ Profile _ Screen?
Subject         = ((AnatomicalPart _ ("with" _ AnatomicalPart)?) / Region)*
Rename          = ("keep" / "continue") _ "to"
CameraMov       = Camera _ ("to" / "with")?
Camera          = Speed? _ ( Dolly / Crane / Zoom / Arc)
Dolly           = "dolly" _ ("in" / "out")?
Crane           = "crane"  _ ("up" / "down") _ ("left"/"right")?
Arc             = "arc"  _ ( "up" / "down")? _ ("clockwise" /"anticlockwise")?
Zoom            = "zoom" _ ("in" / "out")
Speed           = "slow" / "quick" / ("following" Subject)
Transition      = ("cut to" / "dissolve to" / "fade in to")
Cross           = "crosses" _ ("over" / "under") _ Subject _ (Subject)?
Flex            = "flexes"_("partially"/"completely")
Rotate          = "rotates"_("internally"/"externally")
Angle           = "low angle" / "high angle"
Size            = "ECU" / "CU" / "MCU" / "MS" / "MLS" / "FS" / "LS" / "ELS"
Profile         = "ventral" / "dorsal" / "medial" / "lateral" / "ventromedial"
                  / "dorsomedial" / "ventrolateral" / "dorsolateral"
AnatSpecs       = ("Left"/"Right") _ ("proximal" /"distal")?
Screen          = "screen" _ ("top" / "bottom")? _ ("left" / "center"/ "right")?
Plane           = "frontal"/ "transverse" / "sagittal"
Time            = ~r"[0-9]*"i _ ("seconds" / "s" )
String          = ~r"[A-Z 0-9]*"i
Action          = Cross / Flex /  Rotate
Effect          = "appears" / "disappears"
Event           = Subject _ (Action / Effect)
AnatomicalPart  = "Patellar ligament" / "Femur" / "Patella" / "Medial meniscus"
                  / "Fibula" / "Hip bone" /"Tibia"
                  /"Lateral meniscus" /"Articular capsule of left knee joint"
                  / "Posterior cruciate ligament"
Region          = "Knee"
space           = ~r"\s"
_               = (space)*
```

Figure 4.7: ASL Grammar.

Chapter 5

Anatomic scene animation

In the previous chapter, we have defined the input system for the authoring tool. In this chapter, we describe how the scripts are executed to make an animated video with or without narration. We introduce the concept of a style sheet to control the aesthetic of the video. Then we present the Unity application that runs the parsed and translated ASL scripts. Finally, we give the option to recording narrations for the lessons. These narrations are recorded by the teachers and will decide the runtime of the video.

5.1. Style sheets

The descriptive terms of the ASL such as "ventral" or "proximal" need to be converted to numerical values in HFSMs. For example, in this thesis, we specified that the ASL term “high angle” will be translated to a 45 degrees bird’s eye view. This numerical value of 45 is defined in an animation style sheet [85] along with other parameters such as the camera speed. The teachers can edit this stylesheet depending on their preferences, giving them more nuanced control over the video-making process. Some aspects are not included in the ASL grammar, such as time spent on a composition or default time taken for a camera movement. These are global values that change the total run-time of the video and can be edited directly in the animation style sheet.

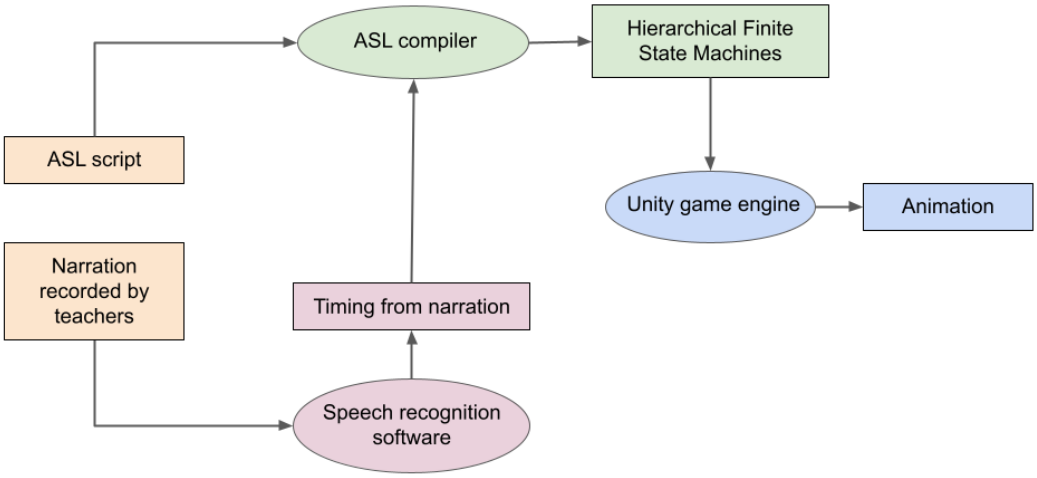


Figure 5.1: Workflow of the ASL Text-to-Movie authoring system

5.2. My Corporis Fabrica and dictionary

All the anatomical terms used in the ASL authoring system are derived from My Corporis Fabrica (MyCF) [81, 82]. It is an ontology connecting the details of anatomical parts with their functions. Ontologies are representations of linked data. They are systems of formalising information in which one can describe the relationship between different data entries that can be exported and used across other domains. For example, an ontology of anatomical data can contain the description of all body parts and their anatomical relation. The information from this ontology can be used in different fields, such as teaching anatomy, creating 3D models for surgical simulations, animation industry for creating realistic human characters, robotics or more. There are many such anatomical ontologies, many of which can be grouped under Open Biomedical Ontologies (OBO) foundry [104]. It consists of multiple ontologies developed in various biological fields such as the Gene Ontology that describes genes and their products and the Coronavirus Infectious Disease Ontology that aims to include all the aspects of coronavirus infection from epidemiology to treatment. The anatomical ontologies in OBO foundry include that of the mouse and drosophila, which are very

commonly used testing systems in biology and a subset of the Foundational Model of Anatomy Ontology (FMA) [94] which is the reference ontology for human anatomy.

The FMA is a comprehensive ontology of the human body from the molecular to the macroscopic scale. It describes the structural organisation of the body. The My Corporis Fabrica ontology takes this anatomical structural description of FMA further by adding the functional relations between the parts. This contribution is helpful in our application as it ensures that we can build 3D models that are anatomically accurate and have a physiological function. And as MyCF is queryable, we can write queries to obtain the list of parts we need to include in our models.

Pseudo code for querying bones in the lower limb
FIND: *Lower limb bones*
WHERE:
Lower limb bones are **Part of** *Skeleton of lower limb* in MyCF
In this we create a *Lower limb bones* variable in which we add all the entities that are listed as being part of the *Skeleton of lower limb* in MyCF.

Pseudo code for querying function
FIND: *Muscles that flex the knee*
WHERE:
Muscles that flex the knee are **Subclass of** *Muscles in MyCF*
Muscles that flex the knee **Participate to** *Flexion of the knee joint in MyCF*
In this we create a *Muscles that flex the knee* variable in which we first add all the entities that are listed under the subclass of*Muscles* in MyCF. Then we find all muscles that participate in *Flexion of the knee joint* in MyCF.

The list of parts is added to the grammar under the Anatomical Parts non-terminal. They can be updated or changed according to the lesson. When transitioning to a new *state* of an HFSM, the application first analyses this list of components named after the MyCF ontology. The names of parts in MyCF need to match the terms of 3D objects in the Zygote model used in Unity player. If the names do not match, we use a dictionary to convert MyCF names to the 3D objects of the zygote model. This conversion is not trivial since the relation is not bijective. Some MyCF objects are divided into subparts represented as one unique entity in the 3D model or vice versa. For example,

“head_of_right_femur” exists in MyCF, but the femur is not subdivided in the zygote model, and only the whole bone can be displayed in this case and in the other case, “Right _gastrocnemius” is listed as one part in MyCF and is divided into its component parts of "Right_gastrocnemius_medial_head", "Right_gastrocnemius_lateral_head" and "Right_Achilles_tendon" in Zygote model. In these cases, we need the dictionary to convert the terms used in ASL scripts to match the terms used to build the 3D model in Unity. The dictionary is also built by using the querying feature of MyCF. In our Gastrocnemius example, we query in MyCf to find all the parts that make up the muscle and then find their names in the Zygote model.

5.3. Unity player

Our collaborators in the Anatoscope startup developed an application using the Unity 3D game engine to generate the desired animation at runtime from the HFSM obtained from the ASL script. The application is thus an interpreter from a specific XML format to 3D videos of anatomy.

The player first builds a 3D space with the zygote model at the center with appropriate lighting. Then it adds the camera to look at the anatomical parts that were mentioned in the script. These parts are listed under a specific anatomy tag in the XML file for each *state* of the HFSM. The placement of the camera is computed automatically using both internal scene data and data from the ASL script converted through the animation style sheet. We use the bounding box of the 3D models to encompass the whole composition when view angles are setup depending on the ASL information. As the shot develops and there is a camera movement, the application computes a path from its previous position to the next one. It adjusts its orientation according to new parameters. The time taken for this movement is set either in the style sheet if there is no narration or by the time set by the narration if there is. If an *animation* tag is present in a XML state, the 3D models get animated following a previously registered animation stored in the application. This animation database has been manually built and can be expanded as per necessity using Anatoscope software.

5.4. ASR Alignment

We then extended the system to include narrated voiceovers for the video lessons. The teachers have a choice to add narrations to their lessons. Suppose they want to have a narrated lesson. In that case, they record it and pass that recording through an Automated Speech Recognition(ASR) software developed by our collaborators at Laboratoire d'Informatique de Grenoble. ASR systems convert audio files and convert them into a sequence of words without a prior transcript of the narration. It works perfectly for us as the teachers can record themselves directly without providing a script for that narration. The input for this system is a .wav audio file. The first step is to perform speech recognition to generate a transcription. Then we use the Kaldi system [88] of forced alignment to align audio with the text. Kaldi is an open-source C++ toolkit for ASR and speech processing. Forced alignment is a process that takes the text from the transcript of audio and finds the time stamp of that words as they occur in the audio segment. At the end of this process, we have the narration transcript used to build the ASL script and time stamps for each spoken word used to retime the videos.

5.5. Retiming

After we get the time for narrated segments and match them to their corresponding ASL counterparts for the video, divide the time between the time spent in that HFSM state and the time taken for the camera movement, if there is any. If there is no narration, the time spent in each state or the delay and the time for camera movement is set by the style sheet's values. It is the usual time taken for the state to progress in the absence of narration. If the time taken for narration is more than twice the sum of the delay and time for camera movements, we calculate the extra time taken by the narration. This extra time is divided and added to the delay and the time taken for camera movements accordingly. We add two-thirds of this extra time to the delay and the remaining one third to the camera movement. We add more to the delay as more narration signifies that the teacher wants to spend more time describing the anatomical parts seen in that state. We don't want to extend the camera movement time as that will interfere visually with the shot as that, unlike the delay, can be seen

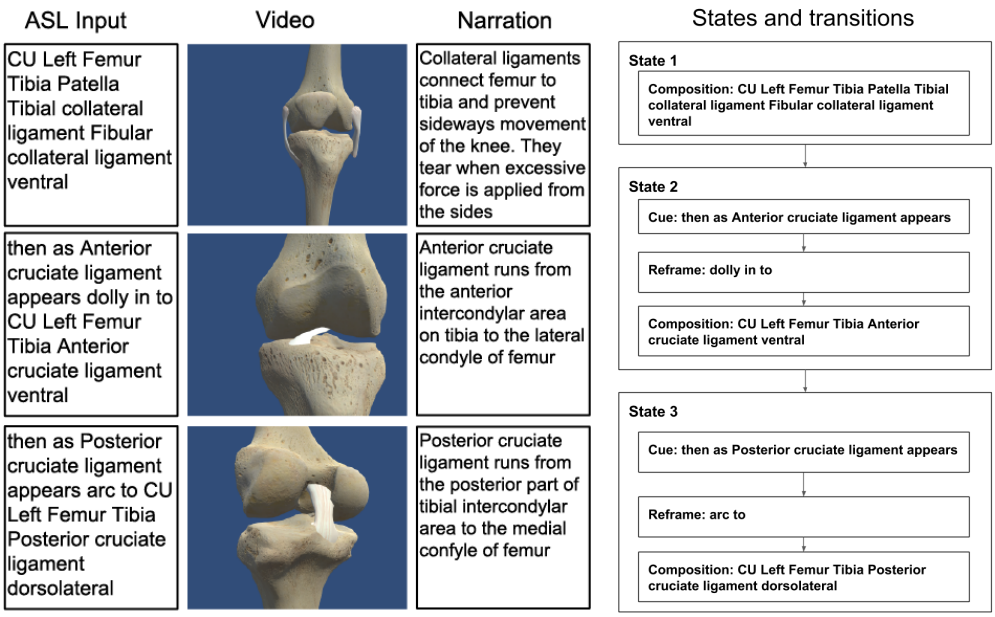


Figure 5.2: Example for the first lesson on bones and ligaments of the knee joint with ASL, corresponding frames from the narrated video and schematic representation of HFSM

in the video. If the narration time is less than the usual shot time, the difference between these two values is divided and subtracted from the delay and camera times. If narration is associated with a lesson, but no part of that narration is linked to a state, then that state will be assigned the delay and camera time from the style sheet.

5.6. Summary

With this, we complete the authoring system. We present an editable style sheet that allows teachers to customise their videos according to their preference and outline the Unity application developed by our collaborators to run the XML files with the Hierarchical Finite State Machines. Finally, we present an option to record voiceovers for these videos. These narrations are processed using Automatic Speech Recognition software to extract time codes for each spoken word. These time codes are used to retime the videos and synchronise the audio to the video.

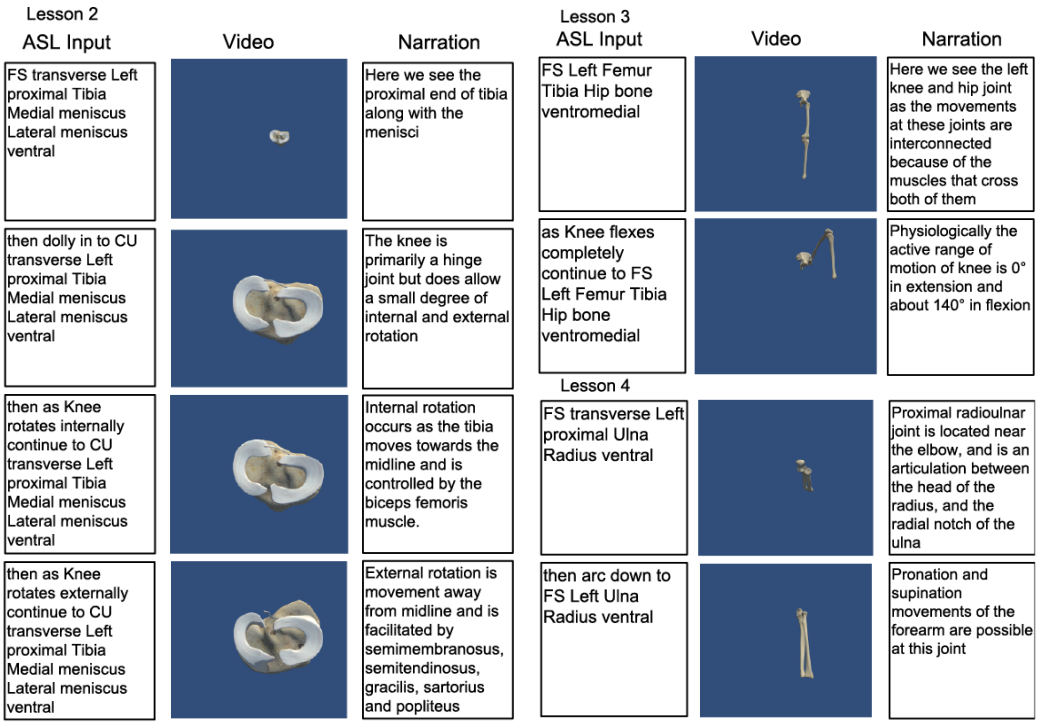


Figure 5.3: Additional examples of ASL scripts and corresponding narrated videos

Chapter 6

Experimental Validation

After we completed the authoring system, we evaluated its ease of use with four anatomy professors. They were new to using 3D models and animation software. We conducted pretest interviews to learn more about their current teaching techniques and assess the need for a video authoring system. Then we introduced the Text-to-Movie using Anatomy Storyboard Language. The teachers were given time to get acquainted with the system and then were asked to make four short lessons using our software. After seeing the output, they were asked to rate the mental workload while creating the lessons and were asked to share their comments. The teachers were enthusiastic about using the system and did not report a very high mental workload. They all expressed interest in spending more time with the system to decrease the mental workload further and create a database of animated videos for their classes.

6.1. Protocol for the Evaluation

The system evaluation was done with four anatomy professors from Laboratoire d'Anatomie Des Alpes Françaises(LADAF) surgical school. They were from varied surgical specialities such as orthopaedics and trauma surgery, pediatric and urologic surgery, cardiothoracic surgery and endocrinology and public health medicine. After they accepted our request to test the system, we organised two sessions to meet in person. The first session was to conduct a short interview of their current teaching

practices and then introduce the Anatomy Storyboard Language. The first session was an hour-long, and after the introduction, the teachers were encouraged to explore the authoring tool by creating their 3D scenes and animations. At the end of the session, they were given the outline for the second session and were asked to record four short audio narrations for lessons on the knee. The four lessons were on the articulations, ligaments, muscles and the movements of the knee joint. Teachers then had to record the audio on any of the devices they were comfortable with and send them to us before the second session. The audio files were then proceeded using Automatic Speech Recognition (ASR) alignment software to extract the time stamps for each spoken word and get the text transcript of the narration.

In the second session, the teachers built the videos based on their recorded narrations. They referred to the transcripts and created the scenes for their lessons one development at a time. Scripts are written in comma-separated values (CSV) files. The first column of the file has the narration and the second column has the ASL sentence fragment that describes the visual elements the teachers want to show for the given narration. Narration for each development or state was added from the transcript. In this way, our authoring system puts 'narration first' as the visual components are built off of the audio parts, and the time spent on the visual states is also based on the duration of the narration for each state.

The second session was two hours long. A summary of the ASL was presented again to make it easier for the teachers to recollect its features. Then, with our assistance, the teachers built lessons, one state at a time. They first referred to and divided the transcript into parts, and for each part, they wrote the ASL fragments. They could visualise their videos and animations immediately and could edit them in real-time as they built them. They could also expand the list of anatomical parts in the grammar to include the parts they were interested in showing.

After the second session, we recorded the videos lessons and added the narrations in Blender, a 3D computer graphics software and sent the completed animated, narrated anatomical lessons on the knee joint back to the professors. Then we organised a short, 15-minute video call to perform the NASA task load test and analysed the results.

6.2. Session 1:Introduction to Anatomy Storyboard Language

6.2.1. Pre-test interviews

The object of the pre-testing interviews was to understand the current system of teaching anatomy in LADAF and assess the teacher's familiarity with video or graphic authoring systems.

1. What is the typical structure of the anatomy classes?
2. What are your prefered tools/ multimedia methods for teaching?
3. How were these videos/other media made?
4. How familiar are you with the animation authoring pipeline?
5. Have you used any authoring systems before?

1. What is the typical structure of the anatomy classes

The general anatomy curriculum in French medical education is divided over two years. The first-year courses have a strength of about 1800 students and are all held online. The first-year courses are similar for students from a wide range of medical and related fields of dentistry, physiotherapy, nursing and others. As the students are from diverse fields, the courses are pretty generalised. They also have in-person teaching sessions in the amphitheatre for smaller groups of about 100 students each. After the students finish the online courses, if they have any questions, they send them to the professors 3 to 4 days before the live teaching sessions. The professors incorporate these questions into their lessons and discuss problem areas in detail with the students. At the end of the first year, examinations enable the students to choose their field. Around 1/10 of the students from the first year pass the exams to join the medical track. These students then start their second year of anatomy courses. The second-year courses are more clinically oriented and focus on problem-based learning by introducing clinical case studies.

2. What are your prefered tools/ multimedia methods for teaching?

During the first-year courses, both online and in the amphitheatre sessions, the preferred teaching method is via PowerPoint slides. For the online classes, these slides have a voice-over from the professors. For the in-person sessions, they reuse the slides without the narration and can show either surgical data or animated videos and talk over them. Previously, the professors would draw the anatomical parts live on the chalkboard as they taught. Still, due to larger class sizes and time constraints, the standard practise now is to use static slides and occasionally animated videos. As the second-year course gets more specialised, the teachers still use slides, but the emphasis shifts to practical sessions with dissections in small groups and one-on-one discussion and introduction to a clinical diagnostic line of enquiry in students.

3. How were these videos/other media made?

The teachers created their slides based on the learning objectives they wanted them to achieve for that lesson. All the teachers have experience in drawing their illustrations. In the qualification examination to become an anatomy professor, they must draw and explain an anatomical concept to a jury. In combination with the rich history in the French medical system of anatomical professors drawing their illustrations, this process ensures that the professors have an extensive database of images to chose from for their slides. They either draw on paper and scan them or draw on graphic tablets. In addition to their illustrations, they also use images from surgical procedures, dissections and 3D models. After they create the slides, they record their narrations using the sound stage in LADAF. These narrations are then processed and synchronised to the slides by an audio engineer. One of the professors records his narrations directly for each slide in Powerpoint due to time constraints. The use of videos and 3D models in teaching is based on personal preferences. As stated earlier, they are more commonly used during the amphitheatre sessions rather than the main online classes. The professor that used videos reported that he uses premade videos from different sources. He has also commissioned one series of videos for his lessons on the pericardium (the double-layered protective sac around the heart). The 3D model was built from computerised tomography (CT) scans of the heart, and then the

labels and camera movements were added to this model. The work was done by an engineer under the direction of the professor and took six months to complete.

4. How familiar are you with the animation authoring pipeline?

All the professors that we interviewed stated that they did not have any experience with animation software or were familiar with creating an animation. One professor was familiar with the use of 3D anatomical models. He had completed a masters course on developing methods of segmentation on 3D knee models. The course focused on simulating osteotomy surgery in 3D space to understand the cause of pain in patients undergoing similar surgeries. But this did not include animations or their use in teaching anatomy.

5. Have you used any authoring systems before?

None of the professors used an authoring system for creating animated videos before. They were interested in testing a system in which the input for the video was a text-based script. This method aligned with the pipeline they had already in place for creating their slides and narration.

6.2.2. Introduction to ASL and Scene building

After the interview, the teachers were given a brief overview of the Anatomy2020 ANR project and this thesis. Then, over the next 45 minutes, they were introduced to ASL grammar using the And/Or tree. Each part of the grammar was explained with examples that the teachers gave themselves. For instance, after the concept of transitions and compositions were presented, the teachers were invited to create a simple shot describing single transitions and the composition of anatomical entities. In this way, the teachers built a few experimental scenes. They were swift to understand the anatomical parts of the language as we used the widely accepted terminology in the medical field. The cinematographic concepts such as shot sizes and camera movements were new to them, which took time to learn. By the end of the session, they were able to build shots with camera movements and animations. The timing of the videos they created was from the values in the style sheet. Then we introduced

the concept of retiming the video based on their narration. The professors were asked to record four short audio clips on four concepts of the knee joint, the articulations, ligaments, musculature and movements. They could record this when convenient, and on any system they liked. They were asked to send the audio clips to us before we conducted the second session.

6.3. Session 2: Narrations and authoring of animated video lessons

6.3.1. Recording narrations and authoring videos in ASL

We analysed the audio files sent by the teachers using the ASR alignment software. The input for this software is the .wav file of the audio narration, and the output is a text file with the time stamp for each spoken word. We then create a text file of this audio transcription which serves as a guide and a script for the teachers to develop their ASL visual directions.

After a short refresher course on ASL in the second session, the teachers started writing ASL parts for the audio script. They chose a part of the audio script and wrote the corresponding ASL description of what they would like to see on screen for that part of the narration. The time spent in each ASL state depends on the number of words from the narration in that state. More words from the narration or a longer narration translated to more time spent in that state. This is a very new concept for teachers. Until now, they are used to recording narrations only for static slides and even then, a sound engineer, in most cases, did the synchronisation of the slides to the audio. Because of this, the speed of the narration was very fast. The teachers did not take many pauses and were not familiar with the average pace for video voiceovers.

As they were building the lessons, they could visualise the video of the ASL script they wrote so far. This real-time viewing and checking of the video was a handy feature for the teachers to edit their lessons on the go. This also highlighted the fact that they could make changes to their video in the future in the same intuitive way.

Lessons	Teacher 1	Teacher 2	Teacher 3	Teacher 4
Articulation	1:42	1:05	0:33	1:22
Ligaments	2:17	2:05	1:03	1:38
Muscles	1:37	0:31	1:23	0:55
Movements	2:00	0:32	1:09	0:17
Total	7:36	4:13	4:08	4:12

Table 6.1: Metrics for the 4 lessons made by the 4 teachers given in minutes and seconds

Adding new items to the list of anatomical parts was also easy. All that was necessary was to find the correct name in My Corporis Fabrica and check if it has the corresponding 3D object in the zygote model.

6.3.2. Final outcome

At the end of the evaluation, the teacher's created four short narrated videos on given topics of the knee. As the teachers were from different specialisations, they had various topics they wanted to emphasise. One teacher wanted to organise the lessons around the patella or the knee cap, while the other wanted to talk more about the knee's ligaments. This was possible with our system as the narrations dictated lessons. The metrics for the lessons are presented in table 6.1. The storyboards for the lessons made by one of the professors are provided at the end of this chapter from Figure 6.3 to 6.9.

6.4. Performance assessment

6.4.1. NASA Taskload test

NASA Task Load Index or NASA-TLX is a commonly used multidimensional assessment tool for measuring subjective mental workload. It was developed in the 1980s by Sandra Hart in the Human Performance Group at NASA's Ames Research centre [44]. Due to its multidimensional approach and generic but effective scaling system, which

can be applied in various fields, NASA-TLX has become a gold standard for quantifying subjective mental workload associated with tasks.

Rubio *et al.* [96] defines mental or cognitive workload as the amount of mental capacity required to complete a given task by an individual. It is a subjective phenomenon, and individual mental workload affects overall task performance. Suppose the mental workload exceeds a subject's capacity. In that case, we start to observe adverse effects in the task performance and outcome such as more errors, decreased attention, higher distraction and inability to reach the result. Measuring an individual subject's workload for specific tasks is a powerful tool for ergonomists and helps them recalibrate their systems. This can be done using NASA-TLX, which can further measure the workload using subscales of task, behaviour and subject-related categories. This can help testers understand the areas of the task the user finds difficult, and they can use this to create more intuitive, productive and comfortable systems for the end-users.

We chose the NASA-TLX assessment tool to understand the mental workload of the anatomy professors when they use our system to create animated, narrated video lessons. This test works well because the teachers do not have any prior experience of using the authoring pipelines. So we cannot compare how our system fares against what is already being used. And comparing our system against their understanding of manually creating their lessons is also not ideal as that system is very long and does not involve the teacher in many of the animation direction steps. In NASA-TLX, we can measure the teacher's workload for the current standalone task of authoring using our system without drawing too much on their past experiences with similar tasks. This way, we get a good quantitative assessment of our system's useability.

To test all the underlying contributing factors that influence a task's workload, NASA-TLX has six dimensions or subscales. They are mental demand, physical demand, temporal demand, performance, effort and frustration. They are divided into three broad types, task-related, behaviour and subject-related. The task-related dimensions are used to measure the objective requirements of the task, such as mental, physical and temporal demands of performing the task. Mental demand is the essential feature and gives an idea about the cognitive capacity required to complete the task. Physical demand is also significant to note as all tasks have a physical component. No matter how trivial the physical nature of the task is, all physical demands require mental

processing. When assessed with the other dimensions, we get a comprehensive view of the nature of the task. The final objective task-related scale is temporal. We ask the subjects if the given time was excessive, enough or too short.

In the behavioural related category, the individual's subjective evaluation of their performance of the task and their effort. For the performance index, it is expected that there will be high bias and low reliability as the subjects rate their performance. Still, when this is paired with other subscales, it is a good indicator of the subjective workload. The effort includes the mental and physics effort exerted to accomplish the task. Finally, in the subject related scale, we have the frustration variable. This measures the psychological impact of the task on the individual. The psychological state of the subject will affect the cognitive abilities during their task performance. This index indicates the subject's comfort level in effectively completing the task in relation to the effort expended and the demands that the task imposed on them. The subscales and their corresponding questions are shown in the form seen in figure 6.1.

The subscales are nearly independent of each other and are general. This makes it ideal for the NASA-TLX test to be used in different domains. The main requirement is a clear and conscience setting of the task. It is also important to note that all tasks are not influenced equally by six subscales. Some of the variables may be irrelevant in specific tasks. Still, this issue is addressed in the weighting procedure, where the user is presented with 15 pairwise comparisons of the six subscales. The user has to chose which of the pair of variables affected the task more. This way, the subscale that was not as important while performing the task will be assigned a lower weight, and their influence on the final score will reflect their contribution to the mental workload.

Following are the steps we took while administering the NASA-TLX:

1. We defined the task on which they were assessed. The task in our case was creating four short animated, narrated video lessons on the anatomy of the knee with emphasis on articulations, ligaments, muscles and movements of the joint using our Text-to-Movie authoring system.

2. We divided the main task into sub-tasks to fully understand the parts and hierarchy of these components in the outcome. The main task of the teachers

authoring video lessons is divided into learning the Anatomy Storyboard Language, using ASL to build short scenarios, recording their narration for the video lessons, writing the ASL to create the visual elements of the video with the narration as the basis of for the lesson.

3. we then briefed the participants on the purpose of the study, gave a detailed description of the task, and outlined the NASA-TLX evaluation tool.

4. The task of authoring the video lessons were performed. We chose not to perform the NASA-TLX test while the task was being performed as it was time-consuming and would affect the task outcome. We performed the assessment post-trial.

5. After the task, we gave the teachers the NASA-TLX rating sheet seen in 6.1. The participants were asked to rate each subscale from 1(very low) to 20(very high). We used software created by Dr Keith Vertanen to perform the test [1].

6. Post rating the teachers performed the weighting procedure. They were presented with the pairwise combinations of the 6 subscales and were asked to select the subscale that contributed more to the workload. The weighted score was calculated by tallying each time a subscale was chosen.

7. The final score was calculated. First, the weighted ratings are calculated by multiplying each rating with its corresponding weighted score. Then the sum of these weighted ratings is divided by the sum of weights which is 15 (there are 15 pairwise weighting steps). This gives the workload score between 0 to 100.

6.4.2. Results

We analysed the six subscales' raw scores from the rating stage and the final workload after the weighing process. We preset the average for each subscale in figure 6.1. This is before the weights have been applied and represent the quantitative subjective rating for each subscale from 1 to 20 (raw scores are given in table 6.2). We find that our system scores well for physical demand, frustration and effort. It also scores favourably for performance. This means that the users found our authoring system to have minimal physical demand and that it did not have a high contribution to the

[1] https://www.keithv.com/software/nasatlx/nasatlx.html

Sub scale	Teacher 1	Teacher 2	Teacher 3	Teacher 4
Mental Demand	14	10	16	13
Physical Demand	2	1	2	2
Temporal Demand	10	13	14	8
Performance	6	5	10	11
Effort	8	3	8	7
Frustration	8	1	2	4

Table 6.2: Ratings for each subscales of NASA Taskload test before weighing

overall mental workload. The physical components of our system were typing an ASL script and recording narrations. The teachers were very familiar with both of these actions. They compared this with other 3D software they were familiar with, in which they had to manipulate the 3D model using an onscreen navigation bar that required a lot of clicking. They found our system physically more intuitive. This was reflected in the low scores for frustration which meant the teachers were not discouraged or annoyed by the system. They did not have to learn any new actions, and for the effort they put in, they achieved the goal they were expecting. The average performance score was 8 out of 20.

The main contributors to the workload of the authoring task were the mental and temporal demands. The teachers had to learn the partonomy and taxonomy of a formalised cinematographic language, ASL. They are not used to thinking of anatomy in terms of the shot sizes or angles. The placement of a camera and its related views and movements were wholly new and took time to be internalised. Another factor was that this was also in English, which is not the native language for the professors. So, they had to think of the visual description of the video with two inbuilt levels of difficulty, a new cinematographic vocabulary and a non-native language. They also specified that they do not have any prior experience in authoring to compare the mental demand of this system. The weighing system helps us in this regard as the teachers can choose which subscale contributed more to the workload, but the comparison is only within the subscales and not with another system. It would be ideal if the teachers used other approaches of creating videos to compare to ours. The temporal demand was high as a consequence of high mental demand. The task

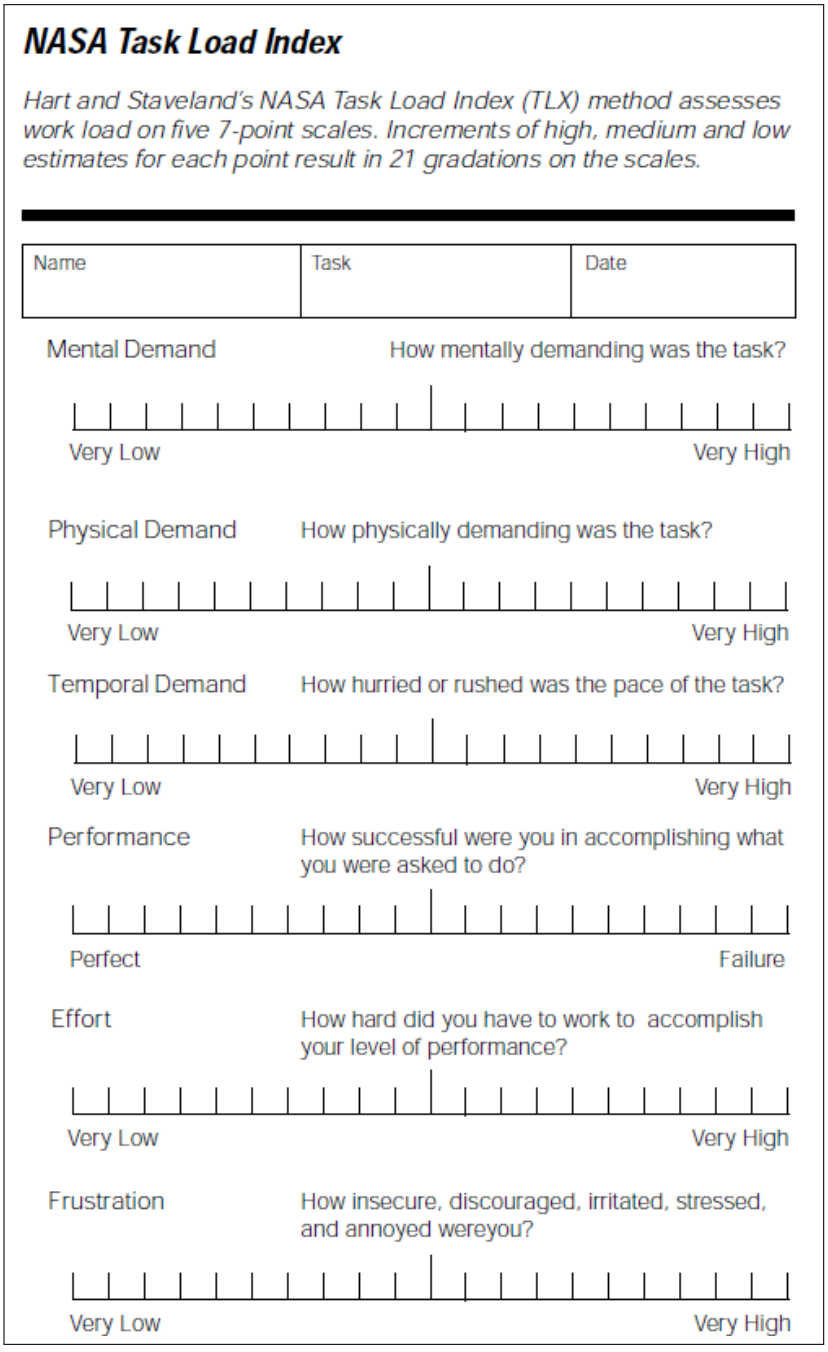


Figure 6.1: NASA-TLX rating scale

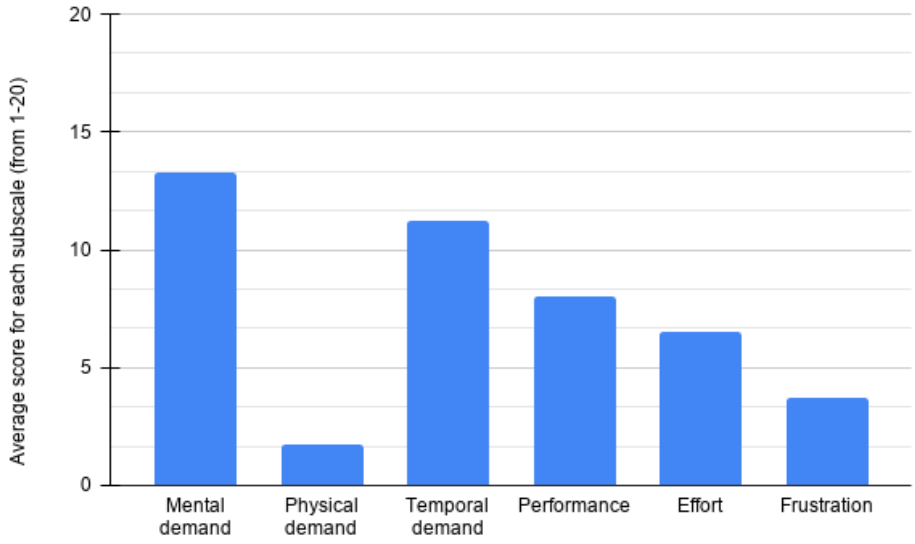


Figure 6.2: Average score for each subscale before weighing

required new concepts that took a lot of trial and error to master. Again the teachers were comparing the time taken to create the videos using our system with the time taken to create slides. They found it challenging to unhook this comparison and to treat this task as a stand-alone entity. Despite these issues, the teachers were very enthusiastic about using the system in the future. They recognised the potential for creative freedom in designing their own digital content and were willing to invest the time and expertise to become more familiar with the authoring tool. They were very enthusiastic about creating their library of videos that can be edited and fine-tuned on a lesson to lesson basis. As the professors involved in our study were at different stages of their teaching careers ranging from veteran teachers with over twenty years of experience to relative newcomers, we had interesting and inclusive feedback. The teachers just starting their teaching tenure were significantly invested in learning and using our system and agreed to further studies to improve the software. They also decided to use the lessons made using our system in their actual classes to get feedback from the students.

6.5. Qualitative feedback and further comments

After the complete sessions of authoring and evaluating the new authoring systems, the teachers expressed interest in further testing the system to get used to this content creation method. The main areas where they faced issues were learning the cinematographic vernacular and estimating the speed of narration. They wanted to use the system more to get used to these new elements. Most of them do not use videos in their regular classes as they did not have the means to create the videos themselves. The teachers stated that there would like to include videos in their lessons if making them were easy. They were very enthusiastic about our authoring system and planned on using it further to create longer lessons.

Concerning the videos themselves, the teachers gave us a list of features they would like to incorporate in the future. These include controlling the opacity of specific parts to see through them, having animated labels of parts that moved as the camera moved and incorporating their current slides and figures into the video lessons. We are already working on these suggestions and chose not to include some of the features into the test version to concentrate more on the ASL authoring aspect. We will incorporate them in the later versions of our system. We plan to conduct the same experiment with the new features and the teachers with prior experience using our authoring system. We believe that will decrease the mental load for the task.

6.6. Summary

We introduced and evaluated a new system for authoring anatomy video lessons with four professors of anatomy from the Grenoble Centre Hospitalier Universitaire (teaching hospital). The teacher had no prior experience with video creating tools. After a short introduction and hands-on tutorial session, the teacher made and manipulated the 3D scenes from the zygote model of the human body using our text-based input system. They learnt the Anatomy Storyboard Language, which combines cinematographic and anatomical terms that describes all the visual elements seen in the video. The teachers learnt to incorporate their audio narrations into the video lessons and found the authoring system intuitive and interesting. The learning curve is a bit steep

as there were new terms and concepts from cinematography that they had to learn and use, but the overall mental workload for the complete authoring task was not high. This workload will only reduce with further use of the system as their familiarity with these new concepts improves. Based on this feedback and the enthusiasm shown by the teachers to learn and use our tool, we will continue to make improvements they suggested, such as including animated labelling and the ability to change the opacity of parts. We focus on the authoring tool as the current thesis aims to teach anatomy pedagogy. The logical next step is to use the lessons made using our system in the anatomy curriculum and get feedback from the students. These validation experiments are being planned for the future and in collaboration with the Anatosocpe start-up team. Given the positive feedback for the authoring tool, we are confident of improving its features and presenting a robust, easy-to-use system for anatomy teachers to use in the future. This work is included in our upcoming article submission in the journal for Artificial Intelligence in Medicine.

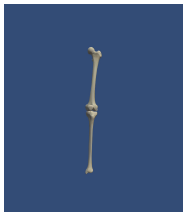
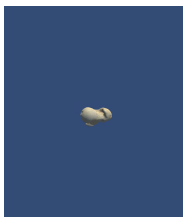
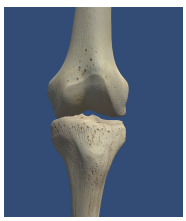
Narration	Anatomy Storyboard Language script	Video screenshots
L'articulation du genou met en rapport trois os: l'épiphyse distale du fémur, l'épiphyse proximale du tibia et enfin la patella	cut to FS Left Femur Tibia Patella ventral	
En réalité le genou comporte trois articulations distinctes	then arc up to FS transverse Left Femur Tibia Patella ventral	
qui sont: l'articulation fémoro-patellaire entre	then dolly in to CU transverse Left Femur Tibia Patella ventral	
la trochlée fémorale et la surface cartilagineuse de la patella	then arc down to CU Left Femur Tibia Patella ventral	
et les deux articulation	then as Patella disappears continue to CU Left Femur Tibia ventral	

Figure 6.3: Storyboard for lesson on articulation - Part 1

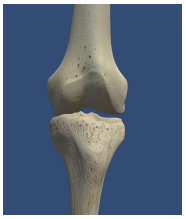
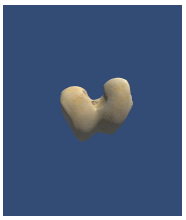
Narration	Anatomy Storyboard Language script	Video screenshots
fémoro-tibiales médiale	then arc clockwise to CU Left Femur Tibia medial	
et latérale	then arc clockwise to CU Left Femur Tibia lateral	
Sur cette vue ventrale d'un genou droit nous mettons en place les deux condyles fémoraux qui sont recouverts de cartilage et qui s'articulent avec les surfaces articulaires de l'épiphyse tibiale proximale pour former les articulations fémoro-tibiales médiale et latérale	then arc clockwise to CU Left Femur Tibia ventral	
Au centre, entre les deux condyles fémoraux il existe une dépression, non recouverte de cartilage appelée fosse ou échancrure inter-condylienne. Au sein de cette fosse se trouvent les deux ligaments croisés du genou.	then as Tibia disappears arc down to CU transverse Left distal Femur ventral	
Ventralement il existe une autre zone recouverte de cartilage: il s'agit de la trochlée fémorale qui va s'articuler avec la surface articulaire de la patella.	then arc up to CU Left Femur ventral	

Figure 6.4: Storyboard for lesson on articulation - Part 2

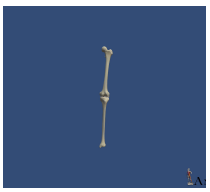
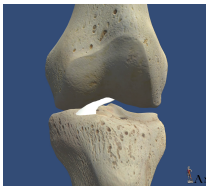
Narration	Anatomy Storyboard Language script	Video screenshots
Les ménisques sont des fibrocartilages situés au centre du genou entre l'extrémité distale du fémur et l'extrémité proximale du tibia.	cut to FS Left Femur Tibia Medial meniscus Lateral meniscus ventral	
Il existe deux ménisques: le ménisque médial qui a une forme de C et le ménisque latéral en forme de O.	then as Femur disappears arc up to FS transverse Left proximal Tibia Medial meniscus Lateral meniscus ventral	
Sur cette vue crâniale de l'épiphyse supérieure du tibia nous pouvons identifier les deux épines tibiales. En avant des épines tibiales se trouve l'insertion distale du ligament croisé antérieur (LCA) au niveau de la surface pré-spinale. En arrière des épines se trouve la surface rétrospinale permettant l'insertion distale des fibres du ligament croisé antérieur (LCP).	then as Medial meniscus Lateral meniscus disappears continue to FS transverse Left proximal Tibia ventral	
	then arc down to FS Left Tibia ventral	
Nous allons maintenant voir le pivot central du genou à savoir le ligament croisé antérieur et postérieur. Le ligament croisé antérieur s'insère en distalité sur la surface pré-spinale du tibia et va en proximalité s'inserer sur la face interne du condyle fémoral latéral. Il a donc quand nous allons de la distalité vers la proximalité une direction ventro-dorsale et médio-latérale.	then as Femur Anterior cruciate ligament appears dolly in to CU Left Femur Tibia Anterior cruciate ligament ventral	

Figure 6.5: Storyboard for lesson on ligaments - Part 1

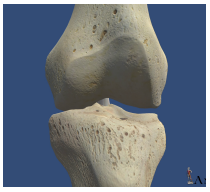
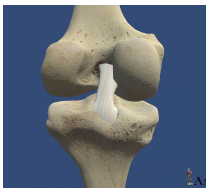
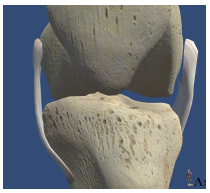
Narration	Anatomy Storyboard Language script	Video screenshots
Le ligament croisé postérieur qui est plus épais et résistant que le LCA	then as Posterior cruciate ligament appears continue to CU Left Femur Tibia Posterior cruciate ligament ventral	
s'insère en distalité sur la surface rétro-spinale du tibia et va en proximalité s'inserer sur la face interne du condyle fémoral médial. Il a donc quand nous allons de la distalité vers la proximalité une direction dorso-ventrale et latéro-médiale.	then arc clockwise to CU Left Femur Tibia Posterior cruciate ligament dorsal	
	then arc clockwise to CU Left Femur Tibia ventral	
Enfin, nous allons terminer par les plans ligamentaires périphériques qui eux sont extra-articulaires.	then as Patella Tibial collateral ligament Fibular collateral ligament Fibula appears continue to CU Left Femur Tibia Patella Tibial collateral ligament Fibular collateral ligament Fibula ventral	
Nous allons commencer par le LCT qui est un ligament résistant et qui s'insère de l'épicondyle médial du fémur en proximalité et qui se termine en distalité le long de la face médiale de la partie proximale du tibia. Il permet de limiter le mouvement de valgus du genou.	then arc clockwise to CU Left Femur Tibia Patella Tibial collateral ligament Fibular collateral ligament Fibula medial	
Enfin nous mettons en place de LCF s'insèrant de l'épicondyle latéral du fémur en proximalité et se terminant en distalité au sommet de la tête de la fibula. Ce ligament permet de contrôler le mouvement de varus du genou.	then arc clockwise to CU Left Femur Tibia Patella Tibial collateral ligament Fibular collateral ligament Fibula lateral	

Figure 6.6: Storyboard for lesson on ligaments - Part 2

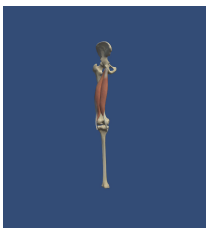
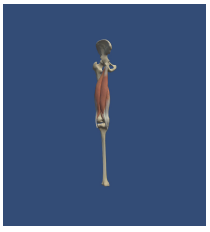
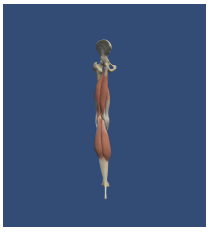
Narration	Anatomy Storyboard Language script	Video screenshots
Nous allons voir l'ensemble des muscles qui composent la fosse poplitée. Sur cette vue dorsale nous allons mettre en place les limites musculaires.	cut to FS Left Femur Tibia Hip bone Patella dorsal	
Le bord proximal et latéral est constitué par le muscle biceps fémoral.	then as Biceps femoris appears continue to FS Left Femur Tibia Hip bone Patella Biceps femoris dorsal	
Le bord proximal et médial est composé des muscles ischio-jambier avec le muscle semi-tendineux	then as Semitendinosus appears continue to FS Left Femur Tibia Hip bone Patella Biceps femoris Semitendinosus dorsal	
et semi-membraneux dans un plan plus profond.	then as Semimembranosus appears continue to FS Left Femur Tibia Hip bone Patella Biceps femoris Semitendinosus Semimembranosus dorsal	
Les deux limites distales sont réalisées par les muscles gastronémiens latéral et médial.	then as Gastrocnemius appears continue to FS Left Femur Tibia Hip bone Patella Biceps femoris Semitendinosus Semimembranosus Gastrocnemius dorsal	

Figure 6.7: Storyboard for lesson on muscles

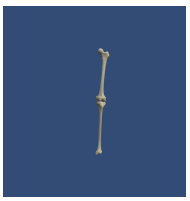
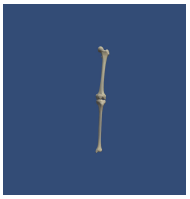
Narration	Anatomy Storyboard Language script	Video screenshots
Sur le plan de la mobilité, le genou comporte deux degrés de liberté à savoir la flexion / extension et il permet également des mouvements de rotation quand le genou est déverrouillé c'est-à-dire lors de la flexion. L'arc de mobilité en flexion-extension part de 0° d'extension	cut to FS Left Femur Tibia Patella ventral	
	then arc clockwise to FS Left Femur Tibia Patella ventromedial	
jusqu'à une flexion de 120 à 140°.	then as Knee flexes completely continue to FS Left Femur Tibia Patella ventromedial	
	then arc anticlockwise to FS Left Femur Tibia Patella ventral	
	then dolly in to CU Left Femur Tibia Patella ventral	

Figure 6.8: Storyboard for lesson on movement - Part 1

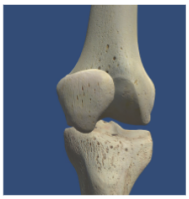
Narration	Anatomy Storyboard Language script	Video screenshots
Concernant les rotations genou en flexion, la rotation externe est de 40°	then as Knee rotates externally continue to CU Left Femur Tibia Patella ventral	
et la rotation interne de 30°.	then as Knee rotates internally continue to CU Left Femur Tibia Patella ventral	

Figure 6.9: Storyboard for lesson on movement - Part 2

Chapter 7

Applications and extensions

Interactive systems are gaining popularity in educational domains [124]. They provide an active learning environment where the students are required to participate and interact with the lessons. They have been successfully used in different fields such as teaching languages via storytelling [76] or highlighting the principles of health care and hygiene to children [33]. Previously, research has been done in developing engaging stories for interactive media, but this was mainly for games in the entertainment sector [11, 105, 110] and in interactive theatre [70, 69]. Research in educational games with an emphasis on learning and problem solving via games [99, 106] is helping the development of serious games. Methods of integrating specific learning objectives into the story of interactive educational games have been explored [73]. Work has also been done to include evaluation within the story arcs of educational games [72]. This evaluation can be used to gamify the learning process, providing the students with an added motivation to improve their scores. This process of teaching enables students to learn with a context and allows for seamless evaluation. Interactive lessons involve the student actively and help them create learning environments for themselves. It helps in experimental and practical training by engaging their decision-making skills. An overview of the different methods of interaction is provided in [121]. In this work, storytelling is stated as a method of exchange that allows the students to adapt to their future professions faster as they play out real-life scenarios while acquiring knowledge. This learning method is an essential feature in medicine, where acquiring knowledge is as crucial as understanding the context where it can be

applied. In this chapter, we present our design for an authoring system that allows teachers to create their own interactive lessons.

7.1. Interactive tools in medicine

The method of learning via case studies is widely used in the medical curriculum at all levels. It helps the students learn and practice within the clinical environment, which improves their decision making and increases their preparedness as future doctors. It is a field that benefits significantly from interactive storytelling. The teachers can create and integrate real-life clinical scenarios in the gameplay [74, 90] and observe the choices students make and help them improve their responses when necessary. [28] present a training tool that helps medical students develop the crucial reasoning skills required for emergency practices. They combine real-world case studies and clinical data with an engaging narrative to make a serious game that enables the user to learn and adapt to working in the trauma bay of a hospital.

Besides training the responsiveness in students to clinical situations, interactive learning can also help them improve their bedside manner of interacting with patients. There has been a push in recent years to develop the field of Narrative Medicine. With the increasing patient load on hospitals and mechanisation of diagnostic protocols, the gap between the doctors and their patients is growing wider. Narrative Medicine uses narrative principles and competencies to connect the patients and physicians better. In this method of patient-doctor interaction, both sides take time to build a narrative around the patient's medical history to humanise the medical experience. First, the doctor acknowledges the patient's suffering and asks detailed questions about their past medical history and current development of illness. Then the doctor combines these to create a cohesive patient-specific narrative that will help both parties understand and manage the disease in a more effective and empathetic manner. Charon describes this in [21] and further detail in the 2016 book on The Principles and Practice of Narrative Medicine [22]. Serous gaming is becoming more and more popular in helping physicians learn the principles of narrative medicine. [19] Generates 3D animated narratives from detailed questionnaires filled out by patients based on their experience in the hospital, especially their interactions with the

physicians. The output can help the doctors analyse how they interact with patients and understand the patients' perspective of the treatment process at different stages concerning their interaction with other health care professionals.

Another field where interactive storytelling is popular in medicine is for patient education. It can either be to give the patients more information about their health condition [23], or to educate at-risk patients about the preventive measures [118, 30] or it can help the caregivers of the patients understand and manage the illness [65, 66].

7.2. Interactive anatomy learning

Anatomy learning is also benefiting from interactivity. [25] presents an overview of some of the work done in implementing game-based learning in anatomy. Most of the research performed in this field focuses on outlining the requirements that anatomy games must have to be educationally relevant in the rigorous medical curriculum [55]. The approach to game-based learning is mainly by adding serious games to the traditional anatomy teaching practices. Most of the interactive tools are digital, but analogue games have also been developed. [3, 46] present board games to learn anatomical concepts in general and, in the case of [46], the anatomy of the liver and the portal venous system. In the digital sector, many commercial and educational tools are being developed to gamify the lessons. The Zygote Body [1] is an anatomically detailed model where the user can explore and create their database of annotated figures. In this way, the students can learn and build their anatomical content, but the input from the teachers is limited. BioDigital [2] is a website that offers detailed anatomical models for over 5000 anatomical parts. The site is aimed at middle school and high school students interested in medical fields. A.D.A.M Interactive Anatomy [3] is an interactive learning tool that allows the teacher to customise the lessons to an extent. Lack of creative control over the content and method of interaction is one of the main limitations of these applications. They lack structured classes that teachers provide. The games are exploratory and help students acquire knowledge, but they

[1] https://www.zygotebody.com/
[2] https://www.biodigital.com/
[3] http://www.adameducation.com/

don't have a planned course based on learning objectives. We aim to design a system that enables teachers to create playthroughs for their interactive anatomy lessons.

7.3. Authoring text-based interactive content

We decided to use text-based interactivity for our system to fit well with case-study oriented learning in clinical anatomy. The final output of our system will be a text-based interactive lesson where the student is given multiple paths to choose from. Their choice will dictate the parts of the lessons they access. This structure is similar to Interactive Fiction(IF). Interactive fiction refers to stories that allow users to interact with it to change the outcome actively. The definition of interactive fiction has since been widely used to refer to text-based narratives focused mainly on puzzle-solving and exploration. Still, it is not limited to this use case. IF is also used to create the narrative for story-rich video games and interactive movies such as Netflix's 2018 movie Black Mirror: Bandersnatch.

Generally, interactive fiction takes text, written in a domain-specific language, as input and use the rules set in a model world to formulate the response to that input [102]. Thus designing interactive fiction has two parts. The first is to create the world and all the elements in it. The second is to specify the rules of the play that dictate how a player can interact with this world. This model world is built before the game starts by the author of the story. It includes the list of characters within the story space, list of objects that the player or non-player characters can interact with, state of these objects(open doors, empty box, etc.), list of interactions that are allowed within the story, location information of the characters and so on. The model world allows for a wide range of actions and affordances for the characters and objects. It is up to the player to select the correct meaningful verbs in their input to perform the action they want. This process requires that the player understands the model world, the story, and its underlying mechanics to proceed in the narrative. [87]. There are several programming languages and applications that can be used to author IF. Inform 7 [4] is one such programing language which used natural language syntax. In this, the fictional world and the rules for interacting with are written in Inform7. For example,

[4] http://inform7.com/

"There is a chair in the room. The chair is empty" describes an object at the starting of the game. Then we can set rules for the chair such as, "If the chair is empty, say "you can sit here"". In the final game, if the player comes across this chair and it is empty, they will be shown the message "you can sit here".

Twine [5] is another open-source authoring system that allows authors to create passages of stories that are linked to each other based on players choices. The author develops the stories in discreet chunks of texts called the passages and then created the choices or links to other passages. For example, "You are in a room with two doors. You can either enter [[the door on the left]] or [[the door on the right]]." This bit of twine code will create two new passages, one called "the door to the left" and the other called "the door to the right". These two passages will have continuations of the story and further choices. This is visually represented in the twine editor with passages as sticky notes and the links as strings that connect the sticky notes. The final software we present is ink by inkle [6]. It is also a text-based IF authoring tool that lets the author write and review the game simultaneously. The game updates live as the author builds the story. The main parts of the ink are knots, diverts and choices. Knots are the main chunks of the story; they can be equated to the passages in twine. Each knot has a name that acts as its identifier. This is used to link sections of the story together. Diverts are directing in which the author can push the story. The diverts lead the narrative from one knot to another. Finally, we have the choices, which the name suggests are the list of choices presented to the player. Based on their choice, the story is diverted to its respective knot. Besides these three concepts, we can also introduce iterable variables that can count the number of times a knot has been visited or keep track of the player's choices. This is valuable in our case for learning as the teacher can observe how their students progress through the lessons. For example, we can see the number of mistakes a student made in figure 7.4. We chose ink as it is an easy to use mark-up language that has a direct plug- in with unity game engine, which we use to build and visualise the videos in our authoring system.

[5] https://twinery.org/

[6] https://www.inklestudios.com/ink/

7.4. Designing an interactive lesson on the knee using ink and ASL

We present a design for an interactive anatomy lesson on the knee in figures 7.1, to 7.4. Introducing interactivity and inbuilt methods of evaluation in lessons are the perspective works of this thesis. We aim to combine the video authoring system with the game authoring capabilities of ink to produce a playable course. In the example that we present the final version of the lesson written by a teacher, in this case, it is me. I build the game in ink but writing knots. The first knot is that of the femur. I write the description that I want the player to see and then add an animated, narrated video of the femur I created using the ASL authoring system. I then present the player with a multiple-choice question. Based on their choice, they get diverted into another knot. In figure 7.2, the player chose the wrong answer, so they are explained that concept with its accompanying video and then are presented with the choice again. When the student gets the answer right, they are shown the number of mistakes they made and can be directed back to repeat the lesson. The teacher has the log of all the choices made by the player within the game and can use this to plan their next courses. In short, the gamified lessons are built part by part in the form of knots. The video lessons are incorporated in the knots. Both the game and the videos can be updated independently of the other, giving the teacher a lot of freedom when editing.

7.5. Summary

In this chapter, we outline our design to author interactive lessons using ink markup language and ASL. The result is an interactive lesson that the student can work through where they learn new concepts and get evaluated on them simultaneously. We believe that this active learning method will benefit the students to have better engagement with their lessons and retain knowledge for a longer time. By giving the teachers the ability to craft the interactive experience, we ensure that the lessons are most effective for the students. Our system can be also be expanded to create videos and interactive lessons in other fields. The main requirement would be to have a library of premade animations. Given this library of labelled animations, the Prose

Storyboard Language can be updated to create content in any field such as physics, geology or embryology. The structure of the grammar remains the same, and so does the structure of the scripts. In this way, our authoring system has a wider reach to include any field with 3D models.

Anatomy of the lower limb

Femur

The femur is the longest bone in the human body. The two femurs converge medially toward the knees, where they articulate with the proximal ends of the tibiae. The angle of convergence of the femora is a major factor in determining the femoral-tibial angle.

It is the only bone in the thigh and serves as the point of attachment for muscles that contibute to the movement of:

Only the hip joint

Only the knee joint

Both hip and the knee joint

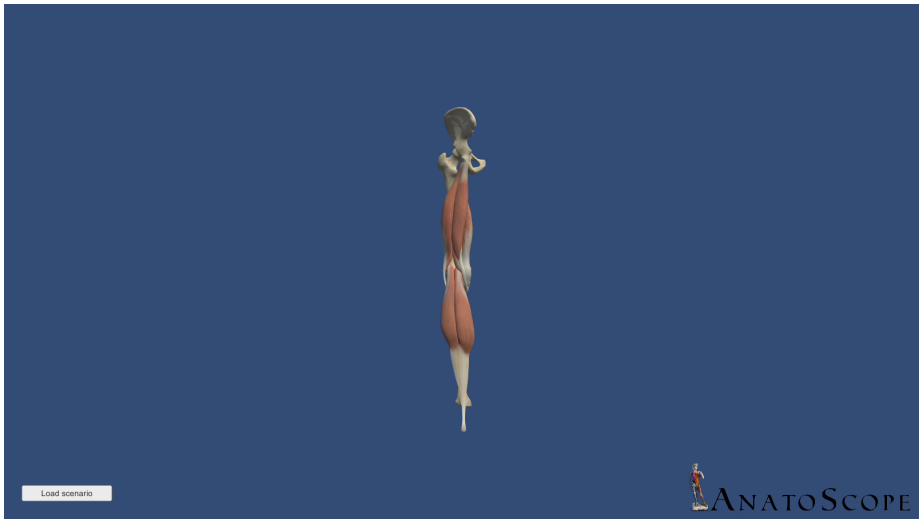
Figure 7.1: Interactive lesson On knee: Part 1

Only the hip joint

Wrong answer

The proximal part of femur (acetabulum) forms a ball and socket joint with the hip bone. Many muscle groups transverse this connection between the axial and appendicular skeleton, notably flexors (such as iliopsoas, rectus femoris,pectineus and sartorius) and extensors such as the hamstrings.

But the long muscles originating from hip bone and femur also cross the knee joint to effect the movement there.



Only the knee joint

Both hip and the knee joint

Figure 7.2: Interactive lesson On knee: Part 2

Only the knee joint

Wrong answer

The distal part of femur flares out to form the lateral and medial condyles. The articular surface of these condyles are interact with the tibial plateau via the articular cartilage and menisci to form a hinge joint with partial rotatory movements. The powerful quadriceps group of muscles cross the knee joint to act as the primary extensors while the hamstring group located posteriorly are responsible for flexion.

But the long muscles originating from hip bone and femur also cross the hip joint to effect the movement there.

Both hip and the knee joint

Both hip and the knee joint

Correct!

Figure 7.3: Interactive lesson On knee: Part 3

You have made some mistakes in this lesson

Number of mistakes made = 2

It would be better to revise before you proceed to the next part

Revise

Proceed anyway

Femur

The femur is the longest bone in the human body. The two femurs converge medially toward the knees, where they articulate with the proximal ends of the tibiae. The angle of convergence of the femora is a major factor in determining the femoral-tibial angle.

It is the only bone in the thigh and serves as the point of attachment for muscles that contibute to the movement of:

Figure 7.4: Interactive lesson On knee: Part 4

Chapter 8

Conclusion

Anatomy is a vital part of medical education, and it is increasingly becoming more digitised. This push towards using digital tools to teach anatomy is necessary because of distance education, increasing class sizes and reduced availability of trained anatomy professors. In the current pipeline, almost all of the animated anatomical teaching content is made by design teams under the instructions of an anatomy teacher. An intuitive authoring system is necessary to give the teachers the creative freedom they need to develop multimedia content for their lectures by themselves. They are most aware of the needs of their students, and they can create animated narrated video lessons keeping learning objectives in mind. We present an easy to use authoring system that takes the text and produces videos. We use text as it's the most commonly used input method for description, and teachers are already used to writing scripts for their lessons; we want to repurpose this skill to feel more ergonomic. Our system removes the need for a design or an animation expert and saves time and money for video production. The system also facilitates easy editing of previous lessons as the lesson is built-in parts, and they can be added or changed without affecting the rest of the lesson.

The text we use is a formalised cinematographic language that has been adapted for anatomy. This language for movies is called the Prose Storyboard Language. It describes all the visual elements on the screen and can be used to direct or annotate movies. It is both human and machine-readable, and we provide the grammar and

parser in Python, which allows the computer to read the PSL sentence written by users and build a syntax tree. This syntax tree is used to create a semantic model of PSL using Petri nets. We propose a method of translating the abstract syntax tree into a graphical Petri net model. Then as a method of evaluation, we manually annotate four scenes from Back to the Future by Robert Zemekis, Rope and North by Northwest by Alfred Hitchcock, and Touch of Evil by Orson Welles. This showcases the versatility and scope of the Prose Storyboard Language.

We then extend this formalisation to apply to anatomy video lessons called the Anatomy Storyboard Language. We adapt the grammar and the parser and simplify the semantic models from Petri nets to linear finite state machines. We build an expandable dictionary of anatomical parts that enables our system to build 3D scenes using the Zygote model of the human body. This dictionary is built using an extensive queriable anatomical ontology called My Coropris Fabrica. We also create editable style sheets that contain the numerical information and directions to translate descriptive elements of ASL, such as profile and camera placement. The script written by teachers in ASL is translated to Hierarchical Finite State Machines using the dictionary and style sheet. A Unity application developed by our collaborators then reads these state machines to produce an animated video.

The next step was to include narration in the animated videos. We do this using the Automatic Speech Recognition (ASR) software developed by our collaborators in Laboratoire d'Informatique de Grenoble. It takes the narration recorded by the teachers in a .wav file as the input and gives the time stamp for each spoken word and the transcript of the voice over. In the workflow of our system, we ask the users to record their narrations before they write the ASL scripts, as this acts as the first step for organising their lesson. They first decide what they would like to mention in the lesson and then make the video. The transcript of the narration helps them in this regard. They can match ASL fragments to their corresponding parts in the narration.

Finally we evaluated the authoring system with four anatomy professors from Laboratoire d'Anatomie Des Alpes Françaises. We first interviewed each to know more about their class structure, the teaching aids they used and their experience with animation and video creation. They reported that the first sessions of their classes were done online, where the students followed a premade lesson. The lessons were

made in the form of a slide show with a recorded voiceover from the teachers. They made their slides, and most drew their illustrations for it. They then recorded their narrations based on a pre-prepared script, and an audio engineer synchronised the audio to the slides. They did not use videos in their presentation as they did not have any methods of making their own. Some reported using videos from YouTube or other sources, but this was not a standard practice. After introducing our system, they were enthusiastic to test it as it gave them creative control over the video-making process. The teachers were first introduced to the project and then a tutorial to authoring scripts in ASL. This was followed by a hands-on practical session where the teachers built their own 3D scenes. They were instructed to prepare recordings for four short lessons on knee anatomy at the end of this. They were asked to send the recording prior to the second session for us to analyse and extract the time stamps for each spoken word and transcript for narration. In the second session, the teachers then created the scripts for their narrations with our help. They built the lesson one state at a time and visualised the video as they wrote.

After the authoring sessions, the teachers were shown the final product and were then given the NASA Taskload test. This was to determine the mental workload they experienced while creating the video lessons using our system. They did not rate high on the cognitive load they exerted, and the main contributor to the workload was the time constraints and new cinematographic concepts they had to learn. We believe that once the teachers are familiar with these terms, they will report a lower workload. In the qualitative feedback, they mentioned that they would like to highlight the anatomical parts in focus, including the feature to control the opacity of parts and introduce a descriptive system in ASL for a pose editor. We are planning on incorporating these in the next iteration of our system. The last feature especially requires further work as editable posing implies that states will no longer be independent of one another. If we set a pose other than the anatomical one and this is mentioned within the lesson, it would mean that state before it must transition to this new manually set pose without a glitch. The second question is that at the end of the state or the animation, should the model go back to the manually set pose or the anatomical pose.

The second field we are extending the thesis into is the implementation of interactive lessons. We currently have the design and the foundation for authoring lessons with

built-in choice based navigation. We would like to extend the current Unity player to include the ink scripts to run the interactivity. We are developing the Petri net semantic model to this end.

We would also like to continue developing the authoring tool to create augmented reality and virtual reality content. This requires a complete rethinking of the camera description. In PSL and ASL, the author sets the camera, and the camera's view then remains the same. But in AR and VR, the camera is the device's point of view being used to view the XR content. Simply stating shot size or the view does not work in these cases. Videos are made keeping the frame of the camera in mind, while XR content is made keeping an explorable 3D world in mind.

Publications and resources

During the course of this thesis we communicated on our experiments with the following articles (Please click on the article to follow the link).

- The Prose Storyboard Language: A Tool for Annotating and Directing Movies - Archive
- Text-to-Movie Authoring of Anatomy Lessons: AIME 2019 – 17th conference on artificial intelligence in medicine

Here are the list of source code links and project pages developed during this thesis:

- Prose Storyboard Language: Link to the results of experimental validation
- Anatomy Storyboard Language:
  - Lessons by Professor 1
  - Lessons by Professor 2
  - Lessons by Professor 3
  - Lessons by Professor 4

Abbreviations

AR	Augmented Reality
ASL	Anatomy Storyboard Language
ASR	Automatic Speech Recognition
FSM	Finite State Machine
HFSM	Hierarchial Finite State Machine
IF	Interactive Fiction
MyCF	My Corporis Fabrica
PSL	Prose Storyboard Language
PTN	Petri Net
SRT	SubRip Subtitle file
TPN	Timed Petri Net
TLX	Task Load Index
VR	Virtual Reality
XML	Extensible Markup Language
XR	Cross Reality or Extended Reality

Bibliography

[1] O. Akerberg, H. Svensson, B. Schulz, and P. Nugues. Carsim: An automatic 3d text-to-scene conversion system applied to road accident reports. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 2*, EACL '03, pages 191–194, 2003. ISBN 1-111-56789-0.

[2] L. Andersen, S. Chang, and M. Felleisen. Super 8 languages for making movies (functional pearl). *Proc. ACM Program. Lang.*, 1(ICFP):30:1–30:29, Aug. 2017. ISSN 2475-1421.

[3] E. G. Anyanwu. Anatomy adventure: A board game for enhancing understanding of anatomy. *Anatomical Sciences Education*, 7(2):153–160, 2014. doi: https://doi.org/10.1002/ase.1389. URL `https://anatomypubs.onlinelibrary.wiley.com/doi/abs/10.1002/ase.1389`.

[4] D. Arijon. *Grammar of the film language*. Silman-James Press ; Distributed by Samuel French Trade, Los Angeles; Hollywood, CA, 1991.

[5] D. Balas, C. Brom, A. Abonyi, and J. Gemrot. Hierarchical petri nets for story plots featuring virtual humans. In *AIIDE*, 2008.

[6] A. Barate, G. Haus, and L. A. Ludovico. Formalisms and interfaces to manipulate music information: The case of music petri nets. In *Proceedings of the 2nd International Conference on Computer-Human Interaction Research and Applications, CHIRA 2018, Seville, Spain, September 19-21, 2018*, pages 81–90. ScitePress, 2018.

[7] F. M. Barreto and S. Julia. Modeling and analysis of video games based on workflow nets and state graphs. CASCON '14, page 106–119, USA, 2014. IBM Corp.

[8] T. Bartindale, A. Sheikh, N. Taylor, P. Wright, and P. Olivier. Storycrate: Tabletop storyboarding for live film production. 05 2012. doi: 10.1145/2207676.2207700.

[9] M. Begleiter. *From Word to Image: Storyboarding and the Filmmaking Process*. Michael Wiese Productions, 2010. ISBN 9781932907674. URL https://books.google.fr/books?id=KYn4PwAACAAJ.

[10] E. M. Bergman. Discussing dissection in anatomy education. *Perspectives on Medical Education*, 4(5):211–213, Oct 2015. ISSN 2212-277X. doi: 10.1007/s40037-015-0207-7. URL https://doi.org/10.1007/s40037-015-0207-7.

[11] K. Bergström. Framing storytelling with games. volume 7069, pages 170–181, 11 2011. ISBN 978-3-642-25288-4. doi: 10.1007/978-3-642-25289-1_19.

[12] L. Blackwell, B. von Konsky, and M. Robey. Petri net script: a visual language for describing action, behaviour and plot. In *Australasian conference on Computer science*, ACSC '01, 2001.

[13] T. Blum, V. Kleeberger, C. Bichlmeier, and N. Navab. mirracle: An augmented reality magic mirror system for anatomy education. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 115–116, March 2012.

[14] D. Bordwell. *On the History of Film Style*. Harvard University Press, 1998.

[15] P. H. C. Braga and I. F. Silveira. Slap: Storyboard language for animation programming. *IEEE Latin America Transactions*, 14(12):4821–4826, Dec 2016.

[16] C. Brom and A. Abonyi. Petri nets for game plot. 3, 01 2006.

[17] S. Calvez, P. Aygalinc, and W. Khansa. P-time petri nets for manufacturing systems with staying time constraints. *IFAC Proceedings Volumes*, 30(6):1487–1492, 1997. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)43571-3. URL https://www.sciencedirect.com/science/

article/pii/S1474667017435713. IFAC Conference on Control of Industrial Systems "Control for the Future of the Youth", Belfort, France, 20-22 May.

[18] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco. Optimal scheduling of human–robot collaborative assembly operations with time petri nets. *IEEE Transactions on Automation Science and Engineering*, 18(1):70–84, 2021. doi: 10.1109/TASE.2019.2932150.

[19] M. Cavazza and F. Charles. Towards interactive narrative medicine. *Studies in health technology and informatics*, 184:59–65, 02 2013. doi: 10.3233/978-1-61499-209-7-59.

[20] C. Chao. Timing multimodal turn-taking for human-robot cooperation. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 309–312, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1467-1. doi: 10.1145/2388676.2388744. URL http://doi.acm.org/10.1145/2388676.2388744.

[21] R. Charon. Narrative MedicineA Model for Empathy, Reflection, Profession, and Trust. *JAMA*, 286(15):1897–1902, 10 2001. ISSN 0098-7484. doi: 10.1001/jama.286.15.1897. URL https://doi.org/10.1001/jama.286.15.1897.

[22] R. Charon, S. DasGupta, N. Hermann, C. Irvine, E. R. Marcus, E. R. Colsn, D. Spencer, and M. Spiegel. *The Principles and Practice of Narrative Medicine*. Oxford University Press, Oxford, UK, 11 2016. ISBN 9780199360222. doi: 10.1093/med/9780199360192.001.0001. URL https://oxfordmedicine.com/view/10.1093/med/9780199360192.001.0001/med-9780199360192.

[23] C.-P. Chiou and Y.-C. Chung. Effectiveness of multimedia interactive patient education on knowledge, uncertainty and decision-making in patients with end-stage renal disease. *Journal of Clinical Nursing*, 21(9-10):1223–1231, 2012. doi: https://doi.org/10.1111/j.1365-2702.2011.03793.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2702.2011.03793.x.

[24] D. B. Christianson, S. E. Anderson, L. wei He, D. H. Salesin, D. S. Weld, and M. F. Cohen. Declarative camera control for automatic cinematography. In *AAAI*, 1996.

[25] D. Chytas, M. Piagkou, and K. Natsis. Outcomes of the implementation of game-based anatomy teaching approaches: An overview. *Morphologie*, 2021. ISSN 1286-0115. doi: https://doi.org/10.1016/j.morpho.2021.02.001. URL https://www.sciencedirect.com/science/article/pii/S1286011521000254.

[26] W. Clifton, A. Damon, E. Nottmeier, and M. Pichelmann. The importance of teaching clinical anatomy in surgical skills education: Spare the patient, use a sim! *Clinical Anatomy*, 33(1):124–127, 2020. doi: https://doi.org/10.1002/ca.23485. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ca.23485.

[27] J. Dai, G. Su, Y. Sun, S. Ye, P. Liao, and Y. Sun. Application of advanced petri net in personalized learning. IC4E '18, page 1–6, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450354851. doi: 10.1145/3183586.3183588. URL https://doi.org/10.1145/3183586.3183588.

[28] M. F. de Menezes Mota, F. L. Pantoja, M. S. Mota, T. de Araujo Guerra Grangeia, M. A. de Carvalho Filho, and A. Santanchè. Analytical design of clinical cases for educational games. In E. van der Spek, S. Göbel, E. Y.-L. Do, E. Clua, and J. Baalsrud Hauge, editors, *Entertainment Computing and Serious Games*, pages 353–365, Cham, 2019. Springer International Publishing. ISBN 978-3-030-34644-7.

[29] Z. Ding, H. Qiu, R. Yang, C. Jiang, and M. Zhou. Interactive-control-model for human–computer interactive system based on petri nets. *IEEE Transactions on Automation Science and Engineering*, 16(4):1800–1813, 2019. doi: 10.1109/TASE.2019.2895507.

[30] A. Dunbar, E. Tai, D. B. Nielsen, S. Shropshire, and L. C. Richardson. Preventing infections during cancer treatment: development of an interactive patient education website. *Clinical journal of oncology nursing*, 18(4):426–431, Aug

2014. ISSN 1538-067X. doi: 10.1188/14.CJON.426-431. URL https://pubmed.ncbi.nlm.nih.gov/25095295. 25095295[pmid].

[31] S. C. Dutilleul, F. Defossez, and P. Bon. Safety requirements and p-time petri nets: A level crossing case study. In *The Proceedings of the Multiconference on "Computational Engineering in Systems Applications"*, volume 2, pages 1118–1123, 2006. doi: 10.1109/CESA.2006.4281811.

[32] M. Estai and S. Bunt. Best teaching practices in anatomy education: A critical review. *Annals of Anatomy - Anatomischer Anzeiger*, 208:151–157, nov 2016.

[33] D. Farrell, P. Kostkova, D. Lecky, and C. Mcnulty. Teaching children hygiene using problem based learning: The story telling approach to games based learning. volume 498, 08 2009.

[34] M. Felleisen, R. B. Findler, M. Flatt, S. Krishnamurthi, E. Barzilay, J. McCarthy, and S. Tobin-Hochstadt. A programmable programming language. *Commun. ACM*, 61(3):62–71, Feb. 2018.

[35] M. Flatt. Creating languages in racket. *Commun. ACM*, 55(1):48–56, Jan. 2012.

[36] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala. Text-based editing of talking-head video. *ACM Trans. Graph.*, 38(4):68:1–68:14, July 2019. ISSN 0730-0301. doi: 10.1145/3306346.3323028. URL http://doi.acm.org/10.1145/3306346.3323028.

[37] D. Friedman and Y. A. Feldman. Automated cinematic reasoning about camera behavior. *Expert Syst. Appl.*, 30(4):694–704, May 2006. ISSN 0957-4174.

[38] Q. Galvane, R. Ronfard, M. Christie, and N. Szilas. Narrative-Driven Camera Control for Cinematic Replay of Computer Games. In *MIG'14 - 7th International Conference on Motion in Games* , pages 109–117, Los Angeles, United States, Nov. 2014. ACM. doi: 10.1145/2668064.2668104. URL https://hal.inria.fr/hal-01067016.

[39] Q. Galvane, M. Christie, C. Lino, and R. Ronfard. Camera-on-rails: Automated Computation of Constrained Camera Paths. In *ACM SIGGRAPH Conference on*

Motion in Games, pages 151–157, Paris, France, Nov. 2015. ACM. doi: 10.1145/2822013.2822025. URL https://hal.inria.fr/hal-01220119.

[40] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F.-l. Tariolle, and P. Guillotel. Directing cinematographic drones. *ACM Trans. Graph.*, 37(3): 34:1–34:18, July 2018. ISSN 0730-0301. doi: 10.1145/3181975. URL http://doi.acm.org/10.1145/3181975.

[41] V. Gandhi and R. Ronfard. A Computational Framework for Vertical Video Editing. In *4th Workshop on Intelligent Camera Control, Cinematography and Editing*, pages 31–37, Zurich, Switzerland, May 2015. Eurographics, Eurographics Association. doi: 10.2312/wiced.20151075. URL https://hal.inria.fr/hal-01160591.

[42] V. Gandhi, R. Ronfard, and M. Gleicher. Multi-Clip Video Editing from a Single Viewpoint. In *CVMP 2014 - European Conference on Visual Media Production*, page Article No. 9, London, United Kingdom, Nov. 2014. ACM. doi: 10.1145/2668904.2668936. URL https://hal.inria.fr/hal-01067093.

[43] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi. Imagine this! scripts to compositions to videos. *CoRR*, abs/1804.03608, 2018. URL http://arxiv.org/abs/1804.03608.

[44] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988. doi: https://doi.org/10.1016/S0166-4115(08)62386-9. URL https://www.sciencedirect.com/science/article/pii/S0166411508623869.

[45] G. Haus and A. Sametti. Scoresynth: a system for the synthesis of music scores based on petri nets and a music algebra. *Computer*, 24(7):56–60, 1991.

[46] R. V. Hill and Z. Nassrallah. A game-based approach to teaching and learning anatomy of the liver and portal venous system. *MedEdPORTAL*, 14, 2018. doi: 10.15766/mep_2374-8265.10696.

[47] N. Hoyek, C. Collet, F. D. Rienzo, M. D. Almeida, and A. Guillot. Effectiveness of three-dimensional digital animation in teaching human anatomy in an authentic classroom context. *Anatomical Sciences Education*, 7(6):430–437, mar 2014.

[48] A. Hulme and G. Strkalj. Videos in anatomy education: History, present usage and future prospects. *International Journal of Morphology*, 35(4):1540–1546, 2017. ISSN 0717-9367. doi: 10.4067/S0717-95022017000401540. Copyright the Author(s) 2017. Version archived for private and non-commercial use with the permission of the author/s and according to publisher conditions. For further rights please contact the publisher.

[49] H. Indzhov, D. Blagoev, and G. Totkov. Executable petri nets: Towards modelling and management of e-learning processes. volume 433, page 38, 01 2009. doi: 10.1145/1731740.1731782.

[50] A. A. Jaffar. Youtube: An emerging tool in anatomy education. *Anatomical Sciences Education*, 5(3):158–164, 2012. doi: https://doi.org/10.1002/ase.1268. URL `https://anatomypubs.onlinelibrary.wiley.com/doi/abs/10.1002/ase.1268`.

[51] N. Jain, P. Youngblood, M. Hasel, and S. Srivastava. An augmented reality tool for learning spatial anatomy on mobile devices. *Clinical Anatomy*, 30(6):736–741, jul 2017.

[52] A. Jhala and R. M. Young. A discourse planning approach to cinematic camera control for narratives in virtual environments. In *AAAI*, 2005.

[53] M. Kapadia, S. Frey, A. Shoulson, R. W. Sumner, and M. Gross. Canvas: Computer-assisted narrative animation synthesis. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 199–209, 2016.

[54] M. K. Khalil, E. M. Abdel Meguid, and I. A. Elkhider. Teaching of anatomical sciences: A blended learning approach. *Clinical Anatomy*, 31(3):323–329, 2018. doi: https://doi.org/10.1002/ca.23052. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/ca.23052`.

[55] A. L. Krassmann, [U+FFFD] M. H. d. Amaral, F. B. Nunes, G. B. Voss, M. C. Zunguze, and L. Tomei, editors. *Handbook of Research on Immersive Digital Games in Educational Environments:*. Advances in Educational Technologies and Instructional Design. IGI Global, 2019. ISBN 9781522557906 9781522557913. doi: 10.4018/978-1-5225-5790-6. URL http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-5790-6.

[56] S. Kucuk, S. Kapakin, and Y. Goktas. Learning anatomy via mobile augmented reality: Effects on achievement and cognitive load. *Anatomical Sciences Education*, 9(5):411–421, 2016.

[57] M. Leake, A. Davis, A. Truong, and M. Agrawala. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.*, 36(4):130:1–130:14, July 2017.

[58] Y.-S. Lee and S.-B. Cho. Dynamic quest plot generation using petri net planning. In *Proceedings of the Workshop at SIGGRAPH Asia*, WASA '12, page 47–52, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450318358. doi: 10.1145/2425296.2425304. URL https://doi.org/10.1145/2425296.2425304.

[59] H. Lin, W.-C. Chang, G. Yee, T. Shih, C.-C. Wang, and H.-C. Yang. Applying petri nets to model scorm learning sequence specification in collaborative learning. volume 1, pages 203– 208 vol.1, 04 2005. ISBN 0-7695-2249-1. doi: 10.1109/AINA.2005.120.

[60] T. D. C. Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communications*, 8: 413–427, 1990.

[61] A. Louarn, M. Christie, and F. Lamarche. Automated staging for virtual cinematography. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games, MIG 2018, Limassol, Cyprus, November 08-10, 2018*, pages 4:1–4:10, 2018.

[62] L. P. Magalhaes, A. B. Raposo, and I. L. Ricarte. Animation modeling with petri nets. *Computers and Graphics*, 22(6):735 – 743, 1998. ISSN 0097-8493. doi:

10.1016/S0097-8493(98)00094-6. URL http://www.sciencedirect.com/science/article/pii/S0097849398000946.

[63] J. Majerník and L. Szerdiová. Preparation of medical students for cadaveric anatomy using multimedia education tools. In *2017 International Conference on Information and Digital Technologies (IDT)*, pages 252–255, 2017. doi: 10.1109/DT.2017.8024305.

[64] D. Markowitz, J. T. K. Jr., A. Shoulson, and N. I. Badler. Intelligent camera control using behavior trees. In *MIG*, pages 156–167, 2011.

[65] S. Marsella, W. Johnson, and C. LaBore. Interactive pedagogical drama. *Proceedings of the International Conference on Autonomous Agents*, 05 2000. doi: 10.1145/336595.337507.

[66] S. C. Marsella, W. L. Johnson, and C. M. Labore. Interactive pedagogical drama for health interventions. In *11th International Conference on Artificial Intelligence in Education*, pages 341–348. IOS Press, 2003.

[67] M. Marti, J. Vieli, W. Witoń, R. Sanghrajka, D. Inversini, D. Wotruba, I. Simo, S. Schriber, M. Kapadia, and M. Gross. Cardinal: Computer assisted authoring of movie scripts. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 509–519, 2018.

[68] T. Marwah, G. Mittal, and V. N. Balasubramanian. IEEE, oct 2017. doi: 10.1109/iccv.2017.159. URL https://doi.org/10.1109/iccv.2017.159.

[69] M. Mateas and A. Stern. Façade: An experiment in building a fully-realized interactive drama. 04 2003.

[70] M. Mateas and A. Stern. Structuring content in the façade interactive drama architecture. In *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'05, page 93–98. AAAI Press, 2005.

[71] G. MejÃa, K. NiÃ±o, C. Montoya, M. A. SÃinchez, J. Palacios, and L. Amodeo. A petri net-based framework for realistic project management and scheduling: An application in animation and videogames. *Computers Operations Research*, 66:190–198, 2016. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2015.

08.011. URL https://www.sciencedirect.com/science/article/pii/S0305054815002087.

[72] A. Molnar and P. Kostkova. Edu-interact: an authoring tool for interactive digital storytelling based games. *Bulletin of the IEEE Technical Committee on Learning Technology*, 18(2/3):10–13, 12 2017. ISSN 2306-0212.

[73] A. Molnar, D. Farrell, and P. Kostova. Who poisoned hugh? - the star framework: Integrating learning objectives with storytelling. In D. Oyarzun, F. Peinado, R. M. Young, A. Elizalde, and G. Méndez, editors, *Interactive Storytelling*, pages 60–71, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34851-8.

[74] N. Morningstar-Kywi and R. E. Kim. Using interactive fiction to teach clinical decision-making in a pharmd curriculum. *Medical science educator*, pages 1–9, Feb 2021. ISSN 2156-8650. doi: 10.1007/s40670-021-01245-7. URL https://pubmed.ncbi.nlm.nih.gov/33643685. 33643685[pmid].

[75] D. Moszkowicz, H. Duboc, C. Dubertret, D. Roux, and F. Bretagnol. Daily medical education for confined students during coronavirus disease 2019 pandemic: A simple videoconference solution. *Clinical Anatomy*, 33(6): 927–928, 2020. doi: https://doi.org/10.1002/ca.23601. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ca.23601.

[76] W. Müller, I. Iurgel, N. Otero, and U. Massler. Teaching english as a second language utilizing authoring tools for interactive digital storytelling. In R. Aylett, M. Y. Lim, S. Louchart, P. Petta, and M. Riedl, editors, *Interactive Storytelling*, pages 222–227, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-16638-9.

[77] S. Natkin and L. Vega. A Petri Net Model for the Analysis of The Ordering of Actions in Computer Games. In *GAME ON 2003*, January 2003. Londres, Octobre 2003.

[78] D. T. Nicholson, C. Chalk, W. R. J. Funnell, and S. J. Daniel. Can virtual reality improve anatomy education? a randomised controlled study of a computer-generated three-dimensional anatomical ear model. *Medical Education*, 40(11): 1081–1087, nov 2006.

[79] M. O'Kane, J. Carthy, and M. Bertolotto. Text-to-scene conversion for accident visualization. In *ACM SIGGRAPH 2004 Posters*, SIGGRAPH '04, 2004.

[80] B. O'Neill, M. O. Riedl, and M. Nitsche. Towards intelligent authoring tools for machinima creation. In *CHI Extended Abstracts*, pages 4639–4644, 2009.

[81] O. Palombi, G. Bousquet, D. Jospin, S. Hassan, L. Reveret, and F. Faure. My Corporis Fabrica: a Unified Ontological, Geometrical and Mechanical View of Human Anatomy. In *3DPH2009 - 2nd Workshop on 3D Physiological Humal*, volume 5903 of *Lecture Notes in Computer Science*, pages 209–219. Springer, Nov 2009.

[82] O. Palombi, F. Ulliana, V. Favier, J.-C. Léon, and M.-C. Rousset. My corporis fabrica: an ontology-based tool for reasoning and querying on complex anatomical models. *Journal of Biomedical Semantics*, 5(1):20, May 2014.

[83] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. To create what you tell: Generating videos from captions. pages 1789–1798, 10 2017. doi: 10.1145/3123266.3127905.

[84] J. Pereira, A. Meri, C. Masdeu, M. Molina-Tomas, and A. Martinez-Carrio. Using videoclips to improve theoretical anatomy teaching. *Eur. J. Anat.*, 8:143–146, 2004.

[85] K. Perlin. Toward interactive narrative. In *Proceedings of the Third International Conference on Virtual Storytelling: Using Virtual Reality Technologies for Storytelling*, ICVS'05, pages 135–147, 2005.

[86] C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, UniversitÃ¤t Hamburg, 1962.

[87] A. Plotkin. Characterizing, if not defining, interactive fiction. In K. Jackson-Mead and J. R. Wheeler, editors, *IF Theory Reader*, pages 59–66. Boston: Transcript On Press, 2011.

[88] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic*

Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011. IEEE Catalog No.: CFP11SRW-USB.

[89] N. Proferes. *Film Directing Fundamentals - See your film before shooting it*. Focal Press, 2008.

[90] D. A. Raines. Med-match: An interactive game to learn medications for clinical practice. *Nursing Education Perspectives*, 41(1), 2020. ISSN 1536-5026. URL https://journals.lww.com/neponline/Fulltext/2020/01000/Med_Match__An_Interactive_Game_to_Learn.28.aspx.

[91] M. Rauterberg, S. Schluep, and M. Fjeld. How to model behavioural and cognitive complexity in human-computer interaction with petri nets. In *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI*, pages 320–325, 1997. doi: 10.1109/ROMAN.1997.647003.

[92] D. V. Rijsselbergen, B. V. D. Keer, M. Verwaest, E. Mannens, and R. V. de Walle. Movie script markup language. In *ACM Symposium on Document Engineering*, pages 161–170, 2009.

[93] R. Ronfard, V. Gandhi, and L. Boiron. The prose storyboard language: A tool for annotating and directing movies. *CoRR*, abs/1508.07593, 2015.

[94] C. Rosse and J. Mejino. A reference ontology for biomedical informatics: The foundational model of anatomy. *Journal of biomedical informatics*, 36:478–500, 01 2004. doi: 10.1016/j.jbi.2003.11.007.

[95] S. Roy, P. Bhakta, D. De, and S. Chakrabarty. Modeling high performance music computing using petri nets. In *Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, pages 678–682, 2014. doi: 10.1109/CIEC.2014.6959176.

[96] S. Rubio, E. DÃaz, J. MartÃn, and J. M. Puente. Evaluation of subjective mental workload: A comparison of swat, nasa-tlx, and workload profile methods. *Applied Psychology*, 53(1):61–86, 2004. doi: https://doi.org/10.1111/j.1464-0597.2004.00161.x. URL https://iaap-journals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1464-0597.2004.00161.x.

[97] B. Salt. *Moving Into Pictures.* Starword, 2006.

[98] B. Salt. *Film Style and Technology: History and Analysis (3 ed.)*. Starword, 2009.

[99] P. Sancho, P. Moreno Ger, R. Fuentes-Fernández, and B. Fernández-Manjón. Adaptive role playing games: An immersive approach for problem based learning. *Educational Technology Society*, 12:110–124, 10 2009.

[100] L. M. Seversky and L. Yin. Real-time automatic 3d scene generation from natural language voice and text descriptions. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, pages 61–64, 2006.

[101] J. Shen, S. Miyazaki, T. Aoki, and H. Yasuda. Intelligent digital filmmaker dmp. In *ICCIMA*, 2003.

[102] E. Short. Interactive fiction. In L. E. Marie-Laure Ryan and B. J. Robertson, editors, *The Johns Hopkins Guide to Digital Media*, pages 289–292. Johns Hopkins University Press, 2014. ISBN 978-1-4214-1223-8.

[103] J. R. Silva and P. M. del Foyo. Timed petri nets. *Petri Nets: Manufacturing and Computer Science, Pawel Pawlewski (ed.)*, pages 359–378, 2012.

[104] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, O. B. I. Consortium, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. H. Scheuermann, N. Shah, P. L. Whetzel, and S. Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, Nov 2007. ISSN 1087-0156. doi: 10.1038/nbt1346. URL `https://pubmed.ncbi.nlm.nih.gov/17989687`. 17989687[pmid].

[105] G. Smith, R. Anderson, B. Kopleck, Z. Lindblad, L. Scott, A. Wardell, J. Whitehead, and M. Mateas. Situating quests: Design patterns for quest and level design in role-playing games. volume 7069, pages 326–329, 11 2011. ISBN 978-3-642-25288-4. doi: 10.1007/978-3-642-25289-1_40.

[106] H. Spires, J. Rowe, B. Mott, and J. Lester. Problem solving and game-based learning: Effects of middle grade students' hypothesis testing strategies on learning outcomes. *Journal of Educational Computing Research*, 44, 06 2011. doi: 10.2190/EC.44.4.e.

[107] K. Stepan, J. Zeiger, S. Hanchuk, A. D. Signore, R. Shrivastava, S. Govindaraj, and A. Iloreta. Immersive virtual reality as a teaching tool for neuroanatomy. *International Forum of Allergy & Rhinology*, 7(10):1006–1013, jul 2017.

[108] A. Stirling and J. Birt. An enriched multimedia ebook application to facilitate learning of anatomy. *Anatomical Sciences Education*, 7(1):19–27, 2014. doi: https://doi.org/10.1002/ase.1373. URL https://anatomypubs.onlinelibrary.wiley.com/doi/abs/10.1002/ase.1373.

[109] M. Stratton and M. Julien. Xtranormal learning for millennials: An innovative tool for group projects. *Journal of Management Education*, 38:259–281, 03 2013. doi: 10.1177/1052562913504923.

[110] N. Szilas, M. Axelrad, and U. Richle. Propositions for innovative forms of digital interactive storytelling based on narrative theories and practices. pages 161–179, 01 2012. ISBN 978-3-642-29049-7. doi: 10.1007/978-3-642-29050-3_15.

[111] R. Thompson and C. Bowen. *Grammar of the Shot*. Focal Press, 2009. ISBN 9780240521213. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Grammar+of+the+shot#0.

[112] R. Thompson and C. Bowen. *Grammar of the Edit*. Focal Press, 2009. ISBN 9780240521206. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Grammar+of+the+edit#0.

[113] D. P. Tim Vernon. The benefits of 3d modelling and animation in medical teaching. *Journal of Audiovisual Media in Medicine*, 25(4):142–148, 2002.

[114] R. B. Trelease. Anatomical informatics: Millennial perspectives on a newer frontier. *The Anatomical Record*, 269(5):224–235, oct 2002.

[115] J. Vogt, M. Haesen, K. Luyten, K. Coninx, and A. Meier. Timisto: A technique to extract usage sequences from storyboards. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '13, pages 113–118, 2013.

[116] T. D. Wilson. *Role of Image and Cognitive Load in Anatomical Multimedia*, pages 237–246. Springer International Publishing, Cham, 2015. ISBN 978-3-319-08930-0. doi: 10.1007/978-3-319-08930-0_27. URL https://doi.org/10.1007/978-3-319-08930-0_27.

[117] A. Winkelmann, S. Hendrix, and C. Kiessling. What do students actually do during a dissection course? first steps towards understanding a complex learning experience. *Academic Medicine*, 82(10), 2007. ISSN 1040-2446. URL https://journals.lww.com/academicmedicine/Fulltext/2007/10000/What_Do_Students_Actually_Do_during_a_Dissection.18.aspx.

[118] K. Winskell, G. Sabben, and C. Obong'o. Interactive narrative in a mobile health behavioral intervention (tumaini): Theoretical grounding and structure of a smartphone game to prevent hiv among young africans. *JMIR Serious Games*, 7(2):e13037, May 2019. ISSN 2291-9279. doi: 10.2196/13037. URL http://games.jmir.org/2019/2/e13037/.

[119] J. Won, K. Lee, C. O'Sullivan, J. K. Hodgins, and J. Lee. Generating and ranking diverse multi-character interactions. *ACM Trans. Graph.*, 33(6), Nov. 2014. ISSN 0730-0301. doi: 10.1145/2661229.2661271. URL https://doi.org/10.1145/2661229.2661271.

[120] H.-Y. Wu, F. Palù, R. Ranon, and M. Christie. Thinking like a director: Film editing patterns for virtual cinematographic storytelling. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(4):81:1–81:22, Oct. 2018. ISSN 1551-6857. doi: 10.1145/3241057. URL http://doi.acm.org/10.1145/3241057.

[121] N. O. Yakovleva and E. V. Yakovlev. Interactive teaching methods in contemporary higher education. *Pacific Science Review*, 16(2):75–80, 2014. ISSN 1229-5450. doi: https://doi.org/10.1016/j.pscr.2014.08.016. URL https://www.sciencedirect.com/science/article/pii/S1229545014000175.

[122] A. Yannopoulos. Directornotation: Artistic and technological system for professional film directing. *J. Comput. Cult. Herit.*, 6(1):2:1–2:34, Apr. 2013.

[123] P. Ye and T. Baldwin. Towards automatic animated storyboarding. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, pages 578–583, 2008.

[124] P. Yuksel, B. Robin, and S. McNeil. Educational uses of digital storytelling all around the world. In M. Koehler and P. Mishra, editors, *Proceedings of Society for Information Technology Teacher Education International Conference 2011*, pages 1264–1271, Nashville, Tennessee, USA, March 2011. Association for the Advancement of Computing in Education (AACE). URL `https://www.learntechlib.org/p/36461`.

[125] R. Zurawski and M. Zhou. Petri nets and industrial applications: A tutorial. *IEEE Transactions on Industrial Electronics*, 41(6):567–583, 1994. doi: 10.1109/41.334574.