Banks, John. "Co-creative technologies." *Co-creating Videogames*. London: Bloomsbury Academic, 2013. 37–64. *Bloomsbury Collections*. Web. 26 Apr. 2023. <<http://dx.doi.org/10.5040/9781472544353.ch-002>>.

Downloaded from Bloomsbury Collections, www.bloomsburycollections.com, 26 April 2023, 07:08 UTC.

Copyright © John Banks 2013. You may share this work for non-commercial purposes only, provided you give attribution to the copyright holder and the publisher, and provide a link to the Creative Commons licence.

2

Co-creative technologies

Introduction

As we have seen in the previous chapter, accounts of co-creative culture by game developers at both Auran and Maxis often revolve around the status of games engine and editing tool technologies. These technologies have a quite active and lively status in their accounts. Exploring the potential of co-creativity required these professionals to develop game engines and software tools that would support these emerging relationships. The designers, producers and programmers also eagerly anticipated the unexpected ways in which gamers may take up these tools to explore the possibilities of modifying and adding on to games such as *Spore*. We need to carefully consider the status of these technologies that openly display their contingency by inviting users to modify them.

During my early discussions with programmers at Auran, my understanding of a game engine was rudimentary. But it was proving difficult to disentangle Auran from technologies such as game engines and editing tools. In the videogames industry, the game engine is the research and development component of game production. As first described to me by Lane and one of Auran's lead programmers, a game engine is a software design environment constituted by a range of sub-systems and functionality that can then be used by designers to implement a particular game design. Game engines render the 3D environments that often characterize contemporary games; they also provide the physics routines that govern the movement and interactions of objects and characters, such as collision detection; they manage sound, artificial intelligence routines, networking and tools. It is this tools layer that is particularly important for co-creative activities such as user-created content as it makes the game and engine accessible so that it can be modified by changing various parameters and by importing new content. The game engine

establishes the limits or parameters within which a design team works; it provides the core foundation software systems for the implementation and development of a particular game design.

In *Unit Operations*, a work that seeks to establish a general framework for videogame criticism, Ian Bogost (2006) notes the significance of game engines in shaping and regulating the expressive possibilities and constraints of videogames. Discussing the game engines that enable first-person shooter titles such as *Doom*, *Quake* and *Unreal Tournament*, Bogost (2006: 54) comments:

The discursivity of games is changed by the capabilities of game engines. The kinds of works, and the nature of these works, have material and functional limitations and capabilities . . . These limitations and capabilities influence the kind of discourse that works can create, the ways they create them, and the ways users interact with them.

Technologies such as game engines are crucial participants then in the making of co-creativity. I have already briefly referred to the *Quake* engine's open architecture that supported and encouraged gamers to develop add-ons and modifications. Similarly, the *Half-Life* 'Counter-Strike' mod was made possible when Valve opened its engine to the end-user community by releasing a software development kit (SDK) and supporting documentation. In the previous chapter we saw Maxis's Lucy Bradshaw, trace co-creativity back to tools such as the *Sim City* Urban Renewal Kit, a toolset released by Maxis for *Simcity 2000* in 1994 that enabled players to customize in-game content such as buildings. Buildings made by fan-artists were then shared on the internet.

In 'The mod industries?' David B. Nieborg and Shenja van der Graaf (2008) consider how game-engine technology shapes the relationship between game developer companies and modders. Focusing on the 'Counter-Strike' modification for the first-person shooter *Half-Life* (Valve 1998), Nieborg and van der Graaf note that these modding practices (in the case of 'Counter-Strike' a total conversion that essentially rewrites the fundamental game mechanics) are part of the rapidly changing relationships in which consumers are now involved directly in the production and distribution of media content.¹ They argue, however, that research on these activities has largely ignored the key role of game engines as proprietary technologies in organizing and shaping these innovation practices (179–80). Drawing on interviews with Valve employees they offer insight into how the relationship between Valve and the modders is organized through the use of Valve's engine technology to follow a particular industrial logic or model (180). Setting out from a similar

observation as Bogost's that engines both structure and constrain game development, including modding activities, Nieborg and van der Graaf argue that 'modding is bound to a large extent by the engine's internal logic, which is operationalized through its proprietary toolkit' (184). They add, 'User-driven innovation through toolkits is structured by the engine and therefore takes place within the set, capital-intensive boundaries of the proprietary technology' (187). In short, Nieborg and van der Graaf conclude that modding activity ends up 'following the industrial logic of game developers' (190) in terms of development and marketing practices.

I agree that we need to pay more attention to how technologies such as engines and tools shape these co-creative relationships; after all that is the starting point for this chapter. However, Nieborg's and van der Graaf's reduction to the category of 'industrial logic' may impede a more fine-grained analysis of these complex dynamics. Very little insight is gained, for example, into how the technologies themselves are shaped by these emergent co-creative practices. How do the developers, for example, respond to what the modders are doing with their tools? We need a more fine-grained account that details the varied views and opinions of professional game developers. I become suspicious when the category and identity of 'game developer' collapses the varied and often competing views of programmers, designers, producers, graphic artists, community managers and so on who contribute to developing games and negotiating the relationships with co-creative players.

Co-creative relations in videogames development are far from a static situation in which modding practices and the associated technologies can be reduced to something like 'an industry logic'. They involve dynamic relationships in which such 'industrial logics' are perhaps themselves unsettled by these emerging practices and technologies. Nieborg and van der Graaf hint at this possibility in the conclusion when they briefly raise the tension between modding and the organizational structures of the game development business (192). They do not fully develop their understanding of the nature of the relationships among game engines and the various practices that they reduce to the category 'industrial logics', such as professional identities, business models, organizational structures, game development processes and so on. Technologies do not simply express these 'industrial logics'; they are also materials and practices that shape and change these various logics. We also need to pay attention to the various participants' (both professional and amateur) understandings of the meanings and usages of these technologies. All of this is much clearer in van der Graaf's (2009) important PhD thesis completed at the London School of Economics and Political Science, 'Designing for mod development:

User creativity as product development strategy on the firm-hosted 3D software platform'. In this study of Linden Lab's *Second Life*, van der Graaf emphasizes the complex dynamics of the firm-user interactions and the growing significance of involving users such as mod communities in knowledge production and innovation practices, which are fundamental for firms such as Linden Labs. Although recognizing the significance of the tool kit and engine technologies, the question of how technologies shape and contribute to these emerging user-participatory relationships is not a primary focus of her PhD study.

Auran's game-engine project

In early 1998, Lane believed open-architecture game-engine technologies and sophisticated but user-friendly editing tools would transform game development. Game engines were becoming more complex as they moved towards modelling 3D environments. Developing an engine that could support a cutting-edge game was becoming a challenging long-term project that required a team of programmers working for at least two years. It was a multimillion dollar undertaking. This created an R&D barrier to entering the game industry. By early 1998, it was increasingly unlikely that a small team of enthusiastic programmers working from their garage would be able to develop a game engine that could then support the design of a commercially competitive game. The alternative was to license a game engine from another developer, and then modify it based on the requirements of the game design. But rapidly escalating licence fees for the high-end engines were again prohibitive.

Auran urgently needed an engine to support its next game project. Lane believed that there was also an opportunity for Auran to become a technology provider, licensing an open-architecture engine to independent game developers seeking to break into the industry. Auran management needed to work out a business model that could effectively take advantage of this opportunity. Building on the success of *Dark Reign*, Auran had developed a significant profile within Australia and the wider international PC games industry. In this environment, the company managed to attract significant investment from both private investors and Australian federal government R&D grants. The aim from 1998 through to 2000 was to raise the local and international profile of Auran by developing an engine technology that could be licensed to other game developers, then to produce a game and secure a publishing deal to cement Auran's position as a top-tier developer, and to promote Auran's technology.

Auran's senior programmers explained to me in early 1998 that the SAGE engine concept involved enabling game designs that were open to end-user content creation. Lane explained:

The extensibility, the customisability, of the engine supports all the add-ons and mods that the fans will create. We headed in this direction with the Tactics Engine for *Dark Reign*; the extensible engine idea grew as we went along. It has now become more important, more significant, as we progress with the new engine. But now it isn't just for the fans. There is an opportunity here, if we get it right, to really open up the game production process. But this will be a big task. It is kind of like trying to build the game engine to do everything. If we want it to be useful for game designers at a generic level, as a general game creation tool or environment, then we need to keep it open so that they can do things with it that we haven't even thought of. It has to be modular and flexible so that designers and their teams can add on to it based on the requirements of their project.

Drawing on discussions and semi-structured interviews with Auran management, programmers, designers and artists, in this chapter I follow the making of SAGE from 1998 through to 2000. This engine (renamed Jet in 2000) also provided the core technologies for the development of Auran's *Trainz* train and rail simulator. This game technology is therefore an important actor in the making of the co-creative networks that I discuss in the following chapters. In engaging with the materiality of objects such as game engines by describing them as 'actants', and thereby attributing to them agency that is often restricted to human actors, I am drawing on ANT. ANT is generally associated with the work of theorists including Bruno Latour, John Law, Annemarie Mol and Michel Callon.

Actor–Network Theory

ANT asks what happens if we refuse the starting-point assumption of a divide between the human and the non-human? By focusing on the process of construction, on the diverse materials that contribute to the making of technical artefacts, Latour questions the assumption of a stable divide between humans and non-humans. For Latour and ANT theorists generally, the social is an outcome – an effect that is made from heterogeneous materials. It is not, in itself, a sufficient explanatory category (Latour 2005: 4–9). As Latour (1999: 109) comments in *Pandora's Hope*, positing an external domain of the social can have an effect of cutting or distancing the complex web of relations that we are seeking to describe and account for

from the 'very things that they connect and assemble'. In *We have Never Been Modern*, Latour (1993: 54) argues that 'before projecting itself on to things society has to be made, built, constructed. And out of what materials could it be built if not out of non-social, nonhuman resources?' (Law 1994: 2; also see Latour 1999: 198–214). Agency or action is never purely human but mediated and translated through artefacts (Latour 1999: 190–3). By pursuing the implications of anti-essentialism to question the human/non-human dualism, ANT also raises the implication that agency is an emergent and distributed quality (Macgregor Wise 1997: 31–2; Latour 2005: 52–8). It is radically externalized through networks rather than residing in the individual human brain or mind. In short, mind is externalized into a collective and emergent intelligence.

Law (2000: 4) describes this unsettling of the human/non-human divide by ANT as a semiotics of the relational. All entities 'achieve their significance by being in relation to other entities. This means that in ANT entities, things, people are not fixed. Nothing that enters into relations has fixed significance or attributes in and of itself'. This approach pursues the implications of a 'radical relationality' or a 'relational materialism' that does not set out from any essential divisions or distinctions. When such distinctions are encountered, they are viewed as complex outcomes or effects, and the task is to determine how such networks of order have been constructed. What are the relations among the heterogeneous materials? How have these relations been articulated and achieved?² As Law suggests, it is 'the analyst's job, at least in part, to explore how those relations – and so the entities that they constitute – are brought into being' (Law 2000: 4). In pursuing this relational project, nothing should be reduced to anything else. ANT often describes this as the principle of non-reduction or irreduction.³

The metaphor of a heterogeneous network is at the core of ANT: society, organizations, enterprises, technologies and agents are effects that emerge from relations among diverse materials. The networks are not so much nouns as verbs, the outcomes of contingent dynamics. The process of bringing materials together to form relatively stable patterns, entities and objects, such as game-engine technologies, game development enterprises or even ethnographic projects, is described as 'heterogeneous engineering'. The ANT approach works to focus our attention on the '*material* character of the networks of the social' in which agents and agency are relational effects, outcomes of ordering struggles enabled and emerging through these heterogeneous networks of materials (his emphases. Law 1994: 100).⁴

In the context of researching videogames, T. L. Taylor (2009: 332) adopts an assemblage theory approach to suggest that games and play are constituted by precisely such interrelations. As she (2009: 332) suggests, 'In the space of interrelations lie the dynamic processes of play'. Yaneva (2009: 277) also

draws on ANT to better understand how designed artefacts enact the social. For Yaneva this concerns,

how objects with their scripts and incorporated programs of action compel and rearticulate new social ties, how the way they are shaped and designed is related to specific ways of enacting the social.

But a problem with such ANT-inspired approaches is that in the effort to describe such relational networks one can end up trying to encompass everything and therefore analytically explaining nothing. What kind of dynamics constitutes these interrelations? In *The Culture of Connectivity: A Critical History of Social Media* (2013), José van Dijck (2013: 26–9) adopts ANT as a theoretical approach for understanding how social media platforms increasingly inform and construct people's sociality. She suggests that ANT usefully allows us to consider 'the co-evolving networks of people and technologies' (26). It allows us to approach platforms such as social media and game engines not simply as artifacts but as a set of relations constituted by humans and non-humans that 'constantly need to be performed; actors of all kinds attribute *meanings* to platforms' (van Dijck 2013: 26). van Dijck, however, argues that although ANT offers a helpful ontological starting point with this refusal to distinguish the social and the technological, it may still be important to make these distinctions for analytical purposes. Here she questions the analytic traction that ANT provides. She argues that such distinctions may be important when we want to understand the changing relationships between technologies and sociality (van Dijck 2013: 26). To overcome this difficulty van Dijck combines perspectives offered by ANT with a political economy (drawing especially on Manuel Castells' work) approach to get at the 'dynamic intricacies of platforms' (van Dijck 2013: 28). ANT enables her to systematically approach platforms as socio-technical constructs while political economy frameworks allow her to highlight platforms as socio-economic structures and consider questions of ownership, governance and business models. van Dijck comments that by combining ANT with political economy, which she acknowledges is an uneasy alliance, she wants to understand the 'coevolution of social media platforms and sociality in the context of a rising culture of connectivity' (2013: 28). van Dijck is raising the right question here. This question is at the very heart of this book – the nature and characteristics of these *co-evolving dynamics* are crucial for understanding co-creativity. I agree that a co-evolutionary approach provides the analytic traction that van Dijck seeks. However, there is also a problem here. van Dijck does not have or propose an evolutionary theory-grounded model of these dynamics that she is trying to understand as co-evolutionary. ANT does not propose such a theory and political economy does not offer

anything like an evolutionary theory-based approach to understanding these dynamics. As van Dijck observes, political economy focuses on structures and not dynamics. I return to this question of evolutionary and co-evolutionary dynamics in the final chapter. For now though I propose that an evolutionary theory-grounded approach to technology and technological change might also provide the analytic traction that ANT is missing. I take up this possibility towards the conclusion of this chapter.

At this stage though I still want to see how far ANT can take us with understanding game engines. Throughout my research relationship with Auran, I certainly noticed the importance of the diverse technical materials that contributed to the making of the growing enterprise, particularly the information technology hardware, infrastructure and software. Without this material there was no Auran. ANT allows us to consider how these technologies participate in the making of assemblages that unsettle the human/non-human divide. Game-engine technology is an important part of the material assemblage that is Auran.

In early 1998 Lane explained to me that Auran was 'back at the drawing board', developing the core engine technology required to create a new game. Lane and other Auran programmers told me in early 1998 that their aim was to expand on the ideas of building a very configurable and extensible game engine. Lane said:

Gamers are modifying games themselves. This is really important, I think, providing the gamers with tools so they can make modifications and add their own content . . . I believe that the extensible, configurable engine approach may well revolutionise the entire game design process, like we've talked about before, involving the gamer, the fans, more in the actual design production process and in creating additional content for the game. Who knows where it will end up taking us.

This definition of the game-engine technology is just a starting point. The status of this artefact is quite elusive. My awareness of the game engine's significance emerged through the dialogues with Lane about Auran's plans for making and managing co-creative relationships with the gamer fans. Lane repeatedly emphasized the importance of the design philosophy and engine architecture that would support the development of these co-creative relationships. At various stages I was shown large, detailed design documents that I was informed 'outlined the architecture of the engine'. I really had no way of assessing these comments or documents. In discussions with Lane and lead programmers, various cues such as body language, hesitations and silences told me that they were working hard to describe a game engine in terms that a layperson with no programming background or knowledge could

follow and understand. They were translating the engine into a framework and a form – reconfiguring it, if you will – to make it useful for me. Was this engine that appeared in the dialogues I had with Lane and the lead programmer the same technology that a programmer would be dealing with in their daily work at Auran? Was SAGE a singular object or was it perhaps multiple? Was I dealing with competing and conflicting perspectives on one engine technology? Or was I encountering multiple engines here?

A game engine is a complex object. It is not singular, but a network of programming modules that can be plugged together to support a particular game design instance. The process of constructing a game engine also requires a lot of material: a network of PCs at which programmers work, sharing their results as the object is constructed; a workspace in which all this material is brought together; and diverse disciplines, professional skills and knowledge, including programming, project management, administration, marketing and financial practices. In my visits to Auran I noticed conference and meeting rooms in which I would glimpse text, flow charts and diagrams that appeared to relate to the game-engine project, but which I had difficulty deciphering and understanding. Intellectual property is also crucial to the network of relations forming around a game engine. As Nieborg and van der Graaf (2008) note, game engines are proprietary technologies and the details of end-user licence agreements carefully regulate the terms and conditions of their use. The questions of who or what owns this software, and the terms and conditions under which others may have access to it, provided a regular topic of discussion at the meetings I sat in on. These intellectual property issues also involved business models that would shape how the engine would be released on to the market and circulate. A game engine is part of and emerges from the complex material organization that is the enterprise network of Auran.

Theorists insisting on the importance of maintaining the boundary between the social and the technological do so because they worry that following ANT's refusal of this divide may give up important territory that has been won through the theory of social construction. In a debate with Latour and Callon, H. M. Collins and Stephen Yearley (1992: 10) argue that ANT's challenge to the separation between the human and the non-human retreats from the position that scientific and technological knowledge is socially constructed. By removing the centrality of the human social world in constructing scientific knowledge and technologies, Collins and Yearley fear that ANT may reinstate the authority of scientists, technicians and engineers. They argue that the sociology of scientific knowledge is 'no longer subservient to the accounts of the work of the scientists and technologists' (321). In conclusion, they suggest that the implications of ANT are far from radical and, if anything, conservative, as 'a poverty of method [makes] it subservient to a prosaic view of science

and technology' (323). Their debate with ANT is driven then by a political concern with the status of the humanities relative to those of the sciences and engineering. This objection would certainly seem to apply to my research situation, in that I was relying on Lane's and the lead programmer's accounts of the game engine in order to construct my research description. In my more recent interviews with Maxis employees I also relied on the programmers' and designers' accounts of the tools and technologies that constitute *Spore*. Everything that I have described so far about Auran's game-engine development has relied on interviews and discussions with Lane and the Auran programmers. But how is it possible to avoid this criticism? Collins and Yearley's point here is that ANT offers *no more* than technologists' account, that Latour and Callon are simply repeating those accounts rather than questioning or challenging them. In short, they are raising questions about the situatedness and politics of research. So am I giving up ground to an entrepreneurial-technologist version of the world that I should instead contest by insisting that technology is shaped by social interests such as those of business or a narrow technicism of software engineers? Further – and this is a point not covered, although I think it is implied, by Collins and Yearley – by just reporting and relying on Lane's and Bradshaw's accounts, is my research participating and colluding in the making of a particular entrepreneurial version of technology projects? What other versions of a game engine and the relations surrounding it are being marginalized here?

Callon and Latour (1992) insist that they are doing more than just relying on the technologists' accounts. The difficulty is that, in empirical fieldwork, when following what scientists and technologists *do*, or even in discussing with them what they do, they 'simultaneously entertain dozens of ontological positions' (352; see also 361). Non-human entities and objects constantly enter the scientists' and technologists' struggles and debates. For researchers to effectively follow and describe these activities, 'it is impossible to take only one of many ontological positions in order to account for the way scientists bring in nonhumans, we the analysts have to entertain the whole range' (Callon and Latour 1992: 353). Annemarie Mol (2002: 33–4) argues that an insistence on maintaining the separation between the social and the technical seeks to assert and protect a domain of knowledge and authority for the social sciences and humanities. The fear is that if the divide is not respected, then other modes of knowledge and disciplines will take over.

Making SAGE

Throughout 1998, I regularly visited Auran, following the development of the game engine technology, SAGE, and the game designs that it would

support. Over this period, Auran moved to a larger inner-city location. Auran's new premises certainly conveyed an impression of a successful and rapidly growing company. Auran was hiring staff needed for the new projects. The company was rapidly growing from a team of some 20–30 to a peak of 60 staff. The programmers' area in particular expanded as the team grew to develop the game engine. Auran employed programmers from Australia, the United Kingdom, the United States, Sweden and Russia. It was an impressive display of people and materials. Off the back of *Dark Reign's* success Auran had attracted considerable investment capital. This included securing an Australian federal government grant of more than AUS\$1 million to support the development of the SAGE game-engine technology.

In mid-1998, on one of my research visits to Auran, Lane introduced me to a young designer recently employed to work on the new game project. The designer explained that Auran had recently licensed the intellectual property to a fantasy role-playing world loosely based on medieval Europe. He was working on a game design that would immerse the gamer in a 3D environment and an epic fantasy narrative. The designer appeared to be excited by the promise of the game engine and the tools that it would make available to create this fantasy environment.

Throughout 1998 I regularly visited Auran and discussed the fantasy game design with the lead designer. However, by late 1998 he appeared to be frustrated with the process. I raised this impression with him and in reply he gestured at a thick document sitting on his desk. He said: 'Look, that is [the game] at the moment, but it isn't a computer game. Well I've got some interesting ideas in the game design docs, but there is no point to it, I don't know what I've got until I see things actually happening on the screen, then I'll have something to show you.' He was frustrated that the SAGE engine and the associated software tools had not yet reached the stage at which he could use them to start implementing the game design:

I really want to get a prototype up and running, then we can see if it works. How does it feel to play, that is the big question. I've got this big, epic story [pointing at the document on his desk], but the important question with [the game's] gameplay will be integrating this story with the experience of playing the game. The story will emerge as you play, in the process of playing. The engine will allow you to do this seamlessly.

He again pointed at the document on his desk: 'All of these systems, how they relate to implementing the design are in there. But I don't actually have the tools or the engine to do anything with and that is the test for me.'

In constructing ethnographic accounts of technological projects, we necessarily rely on what informants tell us. It is often difficult to assess

the status of this information, as we may not have the required technical background. Further, the accounts that we are building and circulating may be contributing to the making of the objects and networks. This is not a disabling condition that can be overcome by deploying a more rigorous methodology or interview technique. It is a mundane and routine part of researching and working in these sites. But is this a simple case of different and even competing perspectives on the SAGE game engine? For the humanities researcher, it offers an opportunity to examine how technologies are reconfiguring relations among producers and consumers. I tended to focus on any comments about the engine's open architecture. For Lane, SAGE concerned business models and intellectual property. For the designer, the engine project presented problems and opportunities for implementing a game design. For the programming team, SAGE was a technical software problem to create 3D environments. They struggled with how objects could interact in these places, how physics would be modelled, and how to achieve all of this within the constraints of limited hardware resources such as memory, graphics capabilities and processor cycles. Is SAGE, then, being constructed from these competing agendas and interests?

I discussed with the designer Lane's views on the extensibility of the engine that aimed to support end-user modifications. He replied: 'Yes, that is an important part of it. They want these systems that I'm describing to you to be open and modifiable.'

He then said there was an issue I should probably be aware of:

This is all big picture. It is all fine in theory, this engine supporting all these systems. It would certainly support some great game designs. But it has to run . . . The programming team are planning on implementing a lot of impressive 3D and graphics effects, using the latest capabilities of the 3D videocards. If it can support all the things they're aiming for it will certainly be impressive. Cutting edge graphics are important. They sell games. But cutting edge 3D graphics come with a cost. My concern comes down to gameplay. As we've discussed before, my approach to game design, to gaming, is that it all comes down really to gameplay. How it feels to play. And this all depends on decent frame rates. And I won't compromise on this. The worry is that all this work going into impressive graphics will result in frame-rate problems and then risk the gameplay.

My interviews with Lane, the designer and the lead programmer were part of an internal debate concerning the relationship between the game-engine development and the game design. The designer was well aware that I would discuss his comments with Lane. I disclosed this before undertaking any interviews with him. He was also aware that all interview transcripts were

provided to Auran senior management for approval before I was permitted to utilize it in my research. Largely to identify any potential commercial in confidence issue. I should note that Auran management on only very few occasions insisted I not use material.

On this particular occasion the designer was clearly trying to communicate an important point to me. It seemed to be important to him that I 'get it'. He continued:

All I'm saying is that this whole generic engine idea, the big vision of an engine that will support all sorts of different game types — the game engine that can do everything — I think there will be a cost to do that. To make a cutting edge game, I really think you need a game engine that is tightly coded and dedicated to the particular type of game that you have in mind. This SAGE, the game engine that can do everything, it is ambitious.

In this discussion with the designer, what SAGE is and will do is an important issue. It was a site of controversy and tension at Auran. In building this account the designer draws from a broad range of materials: hardware specifications and resources, game design principles, business considerations, game-engine technology design and frame rates. How these materials relate and interact is central to what SAGE is. An important point I want to make at this juncture is that the ethnographic account of SAGE that I was compiling is not separate or distinct from the process of building and developing SAGE. The interviews and discussions were yet another material used by those involved, jostling and articulating with frame rates, processor cycles, game design philosophy and business models.

Over the same period (1998–9) I was encountering a very different SAGE in Lane's office. As he described it: 'The engine will radically revolutionise how games are produced and designed.' Lane explained that the important thing for me to understand initially was that 'the methodology guiding SAGE is fundamentally, at the base level, all about redefining what a game is, and what the game design production process involves'. He described a feedback loop in the game production process between programmers and designers. Lane believed that this development loop and the conflicts it generated between designers and programmers often constrained the creativity of game designers, and tended to 'result in a game production process that is too heavily focused on technical code issues rather than the design of an immersive and creative environment that offers good gameplay'. With SAGE, Lane and the team of programmers hoped that the designer and overall design process would be 'freed up from the programmer and code-related issues'. Lane commented that SAGE's open-architecture and tools would potentially shift the game design process from a designer-programmer to a

designer-player relationship: 'It is almost as though the player will collaborate in or even co-produce the game – well to an extent anyway.' This SAGE was a very different object from the one that I encountered in my discussions with the game designer. It aimed to open the game design process to a participatory, co-creative relationship between designers and gamers. Yet the designer at Auran was frustrated that the engine could not yet be used to implement his design to even an early electronic proof-of-concept stage. Are these competing and conflicting representations of the one object? Or is there perhaps more than one SAGE circulating at Auran?

Interviewing Auran: Multiplying SAGES

In June 1999 Lane commented that he thought it was necessary for me to gain a broader perspective on what was going on at Auran by interviewing a cross-section of their staff. He observed that from my discussions with the designer and himself I was probably aware of tensions and conflicts in Auran around the SAGE project. He suggested that I needed an opportunity to put this into context as these kinds of tensions and problems were part of game development. Lane also indicated that he was keen to discuss my views and opinions on what emerged from the discussions. I disclosed to all staff participating in the interviews that the transcripts would be read by senior management, including the CEO. The interviews were conducted across a week in late June 1999 and involved eight Auran staff members, including senior programmers and artists working on SAGE and related game development projects.

Auran's chief operations officer

Auran's COO discussed the importance of SAGE as a open-ended and modular engine technology. He commented, however, that this open-endedness raised problems, as the team – particularly the lead programmer – needed to know where to set the limits:

So we set them a problem, as I put it to the programmers, you have to come up with the answer to a question that I cannot give you. The engine has to be far-reaching. It is a challenge as we want it to be this open, generic technology and yet for it to work at all well, of course there have to be limits to what the systems model.

I asked the COO about his role in managing such an open-ended project. He replied that the most significant challenge was 'setting limits, working out

where the limits are by putting a cap on the features that will be implemented . . . Drawing a line, and saying that will do, it is done, and being able to get that through to people’.

I asked him about his views on the designer’s frustration with the delays in having the engine available to implement the game design:

We’ve got designers and artists who came here to work on making games. They’re eager to be building something. They have the design documents finalised, but everything is on hold because of SAGE. A lot of resources here are going into SAGE at the moment and [the designer] is questioning this. These levels of frustration and even anger are understandable.

He also thought the tensions and conflicts were signs of a growing company heading in a direction that no longer suited some employees.

SAGE team lead programmer

I asked the lead programmer what his views were on the SAGE open-engine concept. He replied:

The game engine that can do everything, yes, when I started Greg [Lane] said we would be building a game engine that could support 3D applications, particularly a fully 3D landscape. They also said it is a game engine that ‘has to do everything’. I knew that was kind of ludicrous because of course any engine technology has to have limits. But my attitude was, ‘sure let’s do it. Let’s see what we can do’.

I asked him whether the amorphous nature of the project presented problems or challenges with managing the team. He replied:

Yes, most definitely. For a while there it was a real mess. Everyone heading off on their own little side projects and directions, doing their own thing around what ‘it has to do everything’ actually is. But now we have refined this down, set some limits on the systems and modules, while at the same time building in the ability to extend on it, keeping it open.

The lead programmer commented that working on SAGE in the lead role had been difficult:

It has been like having two masters. There’s the upper management, the people with the money and the investors. Then there’s the code itself. The

upper management guys may have one idea of how it should turn out and when it should be ready. I understand that, they have their target dates that impact on their break-even points. Some of the guys, the programmers and artists, resent that. They get angry about the pressure or demands from business. But when you think about it, without that, the business, well we wouldn't all be here, doing this, would we? But then there's the code itself, and it is done when it is done. It cannot be half-done as then it just won't run and sometimes, well it does not fit into the business schedules and deadlines. Managing that relationship with the business demands can be awkward. . . This can really frustrate management from a business perspective. They don't like things to be so uncertain.

SAGE team programmer

I asked the programmer for his views of the SAGE project:

The idea of working on a game engine like SAGE really excited me — it was something I wanted to do. The configurable engine idea that carried over from *Dark Reign*, we liked that and saw potential in it . . . The best thing is when one of your ideas, a feature, makes it into the engine. I make a demo of it first, to prove it, that it will work with the engine. Then I need to convince the managers that it should be in there. I've had disappointments with things that didn't get approved and that can be hard. Some of the features I work on at home, for up to ten hours a week or so, even after a 60 hour week here. But I like working the long hours. It isn't even like you're doing a job.

I asked: 'What do you think about the SAGE approach generally.' He replied:

When I first came here and heard the 'game engine that can do everything' idea, well I thought it was too big, too ambitious. Of course I kept that to myself. But now that we have more focus with some limits in place I think it will be good. But the 'do everything' thing is more about marketing. It is hype. Well I think it is.

This programmer spent considerable time describing the challenges of programming the Artificial Intelligence (AI) system for SAGE. He discussed in detail the path-finding AI system that he was working on, checking my notes as we proceeded to ensure that I had got it right. He added: 'Working on this stuff is a childhood dream. We work very long hours, often twelve hours or

more a day. But I would probably go home, sit in front of the computer and do it anyway, it is wonderful work.'

It became apparent that, for this programmer, the interview process provided him with an opportunity to communicate his views about the AI system to Auran's senior management. When I asked directly if he was 'OK with all of the comments that he had made being passed on to Greg', he replied: 'Most certainly, I would like them to know my views.'

The programmers I interviewed seemed to be reasonably happy with the situation at Auran. They all commented on the very long working hours, but then almost without fail would add comments such as: 'But this is a dream job, long hours are part of it and I would be doing it anyway at home. It is not really a job.' The programmers seemed to be strongly invested in the particular technical problems that they were working on. They all questioned the 'game engine that can do everything' descriptor, but considered it more than anything to be a challenge to place some limits and scope on the engine whilst maintaining the extensibility and modularity. Finally, all the programmers interviewed commented that there had been significant delays and problems with the SAGE project that were impacting negatively on the artists and designers. The strongest impression that I took away from my time with the SAGE programming team was that they considered it to be almost a privilege to be working at Auran on games technologies, and that it provided them each with a significant opportunity to work on something that mattered to them. This sense of vocation, of participating in a project that enabled them to work on AI, path-finding, 3D world systems and physics problems compensated for the long hours. I return to this important question of professional identity and work conditions in Chapters 4 and 5.

The graphic artists

The interviews with the artists at Auran were very different in tone from those with the programmers. They were all, to varying degrees, critical of Auran management and expressed strong doubts about the SAGE engine approach. Many of them commented that they had come to Auran on the understanding that they would be working on cutting-edge games. But all added that, for months now, they had not been working on a game design at all. The word 'frustrated' was very common throughout the interviews. They all questioned the generic 'game engine that can do everything' approach driving the SAGE project. The artists and designers felt that if the aim was to develop cutting-edge titles, then this would require a core

engine technology dedicated to a particular game type or genre. One artist commented:

You need a really fast, optimised engine. If that engine is trying to be everything for everyone, well the performance I believe just won't be there. The whole thing is being driven by a programming focus; it is being technology-led and that is a big mistake if you're a games development company.

Another artist commented:

When you work here, you're made to feel that you're very lucky to be in here at all, that only the best work here. To an extent that is so, there are very talented guys here. So you believe it. But then you see all this unnecessary control in the management. They treat us like kids really. Then some of the guys here who questioned the whole game-engine vision, well they are called whingers and told to leave. The atmosphere is of a company trying to be big, a big business and that is not for me. I want to make games.

This frustration came to the fore in my final interview with the lead game designer. He commented:

My concern is the whole project is being driven by the programmers and their objectives. This is about the difference between game design and programming philosophy. It is annoying me, as the decisions they [the programmers] are making now are more about game design, they are intruding into game design areas.

Now don't get me wrong, SAGE is really important and it is an excellent idea. OK, yes SAGE has enormous potential but it runs like a dog and at the moment is hard to program for and design with . . . OK, that's not quite fair. It is better than that and I'm overstating to make a point, to get my point across.

Multiple engines?

The interviews show that there are many different competing and conflicting, even incommensurable, versions of SAGE at Auran. At one point the lead game designer questioned whether the game engine even existed, as he did not have it available to him at the time to build the game design that he was working on. The programmers' versions are a series of technical problems concerning movement systems, AI, 3D world physics and open

modular architecture that they were excited to be working on. Then we have Lane's CEO version of an engine technology that would radically change the way games are designed and developed. He believed that SAGE had the potential to open game development to collaboration between the game design team and gamers. This engine development is also entangled with issues concerning work conditions, authority and relationships between different forms of knowledge. Are these different versions of SAGE, then, conflicting perspectives on a singular thing that contribute to the process of its construction? Is this, after all, a story about competing interests seeking to shape and own this technological object and its role in game development? Is SAGE then socially constructed? For example, we have here quite an interesting story about internal tensions and conflicts at Auran surrounding the development of a technology: designers and artists in conflict with programmers, and both groups questioning the broader corporate strategies and direction. But, as many of the participants at the time pointed out to me, such conflict and disagreement are a common feature of game development. My experience working at Auran on different projects from 2000 through to 2005 and then researching the *Fury* project in 2007 certainly supports this. But by relying on the information provided by Lane and the programmers, am I at all in a position to question or challenge the validity of the versions that they displayed for the ethnographer's gaze? Have I perhaps ended up handing over too much authority to their visions and versions of the world? Are Collins' and Yearley's criticisms of ANT therefore relevant here?

In deploying ANT to briefly follow the making of SAGE, we see that a diverse and heterogeneous network of materials – from designers, programmers and code, to CEOs, intellectual property agreements and PC hardware specifications – went into its construction. SAGE, then, mixes the human and non-human; it involves the production of hybrids. Following the work of Michel Serres, Latour (1993: 50–5) describes them as 'quasi-objects' that so thoroughly mix the human and non-human that it makes little sense to describe them as socially constructed or simply an outcome of human agency and practice. The materials constituting SAGE mediates the various relationships and activities that are foregrounded in the interview extracts – from design and programming practice to workplace relationships (Yaneva 2009).

SAGE is an object that is 'simultaneously real, discursive and social'; it belongs to both the collective and discourse (Latour 1993: 64). These shifting objects with their variable ontologies are not just intermediaries, linking separate and independent things or subjects. SAGE is not only a device or exchange point for coordinating the respective interests and agendas of CEOs, programmers, videocards, AI and 3D world physics modelling systems, processor cycles, 3D art objects, designers and their

game designs, ethnographers and their research projects. More significantly, Latour's approach suggests that they are mediators – 'actors endowed with the capacity to translate what they transport, to redefine it, redeploy it, and also to betray it' (Latour 1993: 81, 2005: 37–42). This is an important uptake of the fundamental ANT assertion that objects are relational effects. This commitment to a relational or semiotic materialism not only focuses our attention on the nature of the diverse materials that constitute a network from which objects emerge; it works to foreground the strategies that enact and extend the network, holding it together as a particular pattern (Law 1994: 23–5). John Law (Law 2000b: 2) describes these coordinating strategies as 'modes of ordering'. Agency and agents are relational effects, outcomes of ordering struggles in which agency is far from secure and centred, and is very much a 'precarious achievement' (Law 1994: 100–1).⁵ In a detailed account of the work of translation in network-building, Latour (1987: 103–6) observes that in enlisting others, both human and non-human, the network builder works to strengthen and extend the network: it is a collective process. But the action of enlistment is never neutral; the object is modified, transformed and reconfigured as it moves through and along the network. The concept of translation articulates the problem and task of enrolling others so that they participate in the work of construction while controlling their actions to ensure that the network retains coherence. There is a tension here. The very process of enrolling others carries the risk of losing control of the network, and of its materials and objects heading off in unexpected directions and taking on different configurations and shapes. Translation describes the tactics employed to tie others' interests to the project of extending the network while maintaining control and coherence. In short, translation strategies are deployed to ensure that the objects and networks remain yours (Latour 1987: 108–21).

SAGE changes as it moves through the network: a CEO's business model can also be a tool for implementing a game design, a physics modelling machine, an AI system, and a problem in the coordination of Auran staff and resources. For example, Lane's entrepreneurial stories are an attempt to impose an order on these materials, that of the go-getting development studio from Australia pursuing opportunities in the United States, making good and securing the required publishing deal and investment capital and then releasing a successful game title. This tale of the heroic entrepreneur embraces a particular way of organizing Auran as a collective of heterogeneous materials. It performs and enacts a particular type of agency and overlooks others.

In building these accounts about the Auran enterprise and the objects such as SAGE that surround and emerge from it, ANT asks us to consider how we participate in the construction of these networks and objects.

The methodological problem then does not only involve how ethnographic objects are constructed as objects of knowledge, but also how they relate to and intervene in other versions of the networks and objects that are being studied. But, in making this point, it is also important to maintain the ANT understanding that the ontology of these objects is variable and multiple. Mediators such as SAGE are far from singular or settled; these quasi-objects 'are collective because they attach us to one another, because they circulate in our hands and define our social bond by their very circulation. They are discursive, however; they are narrated, historical, passionate, and peopled with actants of autonomous forms' (89).

John Law and Annemarie Mol have pursued the implications and potential of this thinking about variable ontologies to question the network logics and metaphors of ANT. In *The Body Multiple: Ontology in Medical Practice* Mol (2002) argues that her ethnography of medical practices surrounding the disease atherosclerosis is not a study of how 'medicine *knows* its objects' (vii), but rather concerns how medicine enacts realities. These objects are multiple and need to be coordinated (vii–viii). She carefully describes how atherosclerosis is enacted through the patient-doctor interview, through the technical practices and diagnostic tools in a vascular laboratory, and in the practices of radiologists and pathologists, as well as in the work of vascular surgeons and the materiality of a haematologist's research laboratory. By drawing on interviews with the medical practitioners, patients, scientists and laboratory technicians, and on descriptions of the laboratories, technical equipment and general practitioners' offices, is Mol simply providing us with different perspectives of a singular object? She argues that by focusing on these diverse practices and heterogeneous materials through which atherosclerosis is manipulated and made, the object multiplies. We are not dealing with a single object waiting to be discovered or disclosed by these various knowledge practices. It is not so much a matter of knowing as it is of doing: we find objects coming into being through these practices. The atherosclerosis that is enacted and made visible beneath the microscope in the department of pathology is very different from that in the general practitioner's office or the outpatient clinic. The techniques, equipment and practices that make the disease visible, knowable, treatable and manageable in these places often exclude one another. Mol's ethnographic methodology seeks to attend to this multiplicity of the object. She then asks how the different versions of atherosclerosis are coordinated, as they need to be: diagnosing and treating the disease requires coordination and cooperation across divergent and at times even conflicting versions of the object:

The practices of enacting clinical atherosclerosis and pathological atherosclerosis *exclude* one another. The first requires a patient who

complains about pain in his legs. And the second requires a cross section of an artery visible under the microscope. These exigencies are incompatible, at least: they cannot be realized simultaneously . . . The incompatibility is a practical matter . . . In the outpatient clinic and in the department of pathology, atherosclerosis is *done* differently. (Her emphases, 35–6)

Now this understanding of the multiplicity of objects marks a break or development from the work of Latour and the ANT of the 1980s and 1990s. The focus now is not so much with the work to construct or stabilize the identities of networks and objects and the difficulties encountered in challenging these materially entrenched networks. Instead, Mol's work suggests that these objects are fragmented and multiple, spread across sites. Coordination across and between these multiple versions of an object becomes an achievement that requires a continuing effort (Mol 2002: 42–3; also see Law 2002: 32–7). Mol describes in detail the resources and work required to coordinate the multiple versions of atherosclerosis that emerge in clinical practice. She examines the establishment of hierarchies, the privileging of one version over another. She is interested in how, in practice, these competing, incommensurable versions of atherosclerosis are organized and coordinated through the health care system (43–8). Atherosclerosis multiplies as Mol follows it across different sites, from the outpatient clinic to the department of pathology. She argues that different ways of enacting the disease coexist across the sites: 'The fact that there is multiplicity stays the same, in every site, on every scale' (50).

We have seen this multiplicity in action at Auran with SAGE and at Maxis in the various accounts of the editing tools for *Spore*. Mol maintains, however, that such a multiplicity is not necessarily fragmented. There are connections and coordinations between versions. As she puts it: 'Atherosclerosis enacted is more than one – but less than many. The body multiple is not fragmented. Even if it is multiple, it also hangs together' (55). Mol describes the forms of coordination that occur across the hospital. Equipment and technologies play an important role in this. A hierarchy may be established and institutionalized between laboratory findings and the patients' 'complaints' (63). We have also seen this work of coordination at Auran as the programmers' SAGE, a problem in modular architecture and a generic, open game engine, was prioritized over the designer's version of a game engine. As Mol comments: 'One of them wins. The other is discarded. Thus a single patient ends up with a single atherosclerosis' (66). In the case of Auran, then, do we end up with a single SAGE – the generic, open-architecture game engine? But is it that simple? Of more interest is how incoherencies between different objects are added or combined as a powerful way of creating singularity as a composite object (67, 69–71). The point here is the various contestations

and misunderstandings among the various participants are not an obstacle to performing and making the game engine but instead are a condition of possibility. In the context of an ethnography of a new media start-up, David Stark (2009: 108–11) describes this dynamic as a ‘discursive pragmatism’ in which collaboration and coordination *requires* such unsettling frictions. He refers to this as ‘the friction of competing performance principles’ and ‘the friction of multiple evaluative frames’ (109). I return to Stark’s work in some detail in the final chapter, as these dynamics are crucial to my understanding of co-creativity.

The versions of SAGE that I encountered at Auran are not a fragmented pluralism. Although often incommensurable, they do hang together in a performance of a software engine technology, business model, complex, composite object. Mol describes the process of enacting atherosclerosis:

Clinical findings, pressure measurement, social inquiries, duplex outcomes, and angiographic images are all brought together in the patient’s file . . . This, then, is what I would like the term *multiple* to convey: that there is manyfoldedness, but not pluralism. In the hospital *the body* (singular) is *multiple* (many). The drawing together of a diversity of objects that go by a single name involves various modes of coordination. (Her emphases, 84)

Incommensurabilities then are not necessarily obstacles to the successful performance of a game-engine development project. Nor are tensions and conflicts between corporate entrepreneurial agendas and academic humanities research or between the market-driven incentives and motivations of game development firms and the non-market values and motivations of gamers obstacles to the successful enactment of co-creative relationships for mutual benefit. Mol carefully argues that the different ways of enacting the reality of the disease need not be reduced to a contradiction. The incommensurabilities that she describes are not a flaw that needs to be mended or overcome. I will return to this challenge of coordinating apparently incommensurable materials and actors as it is a central dilemma for co-creativity. The challenge then becomes understanding the specific conditions in which such relationships can be constituted for mutual benefit. The know-how and communicative craft skills to work comfortably across incommensurabilities, holding terms together without reducing them to a contradiction – are a much-needed resource for negotiating the complexities of these emerging relationships.

In pursuing the implications of this approach, Mol takes up Susan Leigh Star’s theory of boundary objects. Here she is interested in how the movement and circulation of objects may blur and reconfigure the boundaries between social domains and worlds. Star argues that, although different fields of social practice have their own codes, habits, instruments and practices,

they may share a boundary object. The different fields will then attach different meanings and values to this object. We have seen, for example, that a game-engine technology can be very different to programmers, a game designer, graphic artists and a CEO. Mol (2002: 138) comments that 'as long as nobody stresses these differences, the boundary object doesn't seem to be two or three different objects. It remains fuzzy enough to absorb the possible tensions . . . Thus, it facilitates collaboration across boundaries and thereby makes these boundaries less absolute. It blurs them. Blurring boundaries is a way of contesting them'.⁶ However, Mol questions the reliance of this understanding of the boundary object on 'the idea that there are different regions. Adjacent to one another. With a lot of fuzziness between them, but separable, separate, all the same' (139). For Mol, this conception of the boundary object cannot quite get at the uncertainty or undecidability in the movements between difference and similarity. Instead, she uses the term 'fluid' to describe 'transient situations where there is *both* difference *and* similarity' (142). This image works to stress that transitions are not always clearly defined; we cannot always be sure where the boundaries are. They move and are reconfigured.⁷

This is a crucial argument in Mol's work and marks her departure from – or perhaps more correctly reworking of – the network metaphor and logics of ANT (142–3). Different versions of objects are not connected in a seamless network; they blur and flow over and around their others. But the point being made here is not just one about blurring divides. More importantly, it concerns the 'coexistence of what is markedly different'. Mol is articulating different forms and strategies of coexistence, including incorporation and inclusion (143). Instead of things and objects that we may want to oppose to one other taking up settled positions in a Euclidean conception of space such as a network, Mol asks us to consider how they may also '*depend* on one another' (145). She suggests, 'what is opposed may also collaborate' (145). Mol describes these patterns and relations in the following terms: 'Coexistence side by side, mutual inclusion, inclusion in tension, interference: the relations between objects enacted are complex. Ontology-in-practice comes with objects that do not so much cohere as assemble' (150).⁸

The call to participate

Law (2002: 88) raises the concern, however, that this cultural strategy of emphasizing partial connections, discontinuities and multiplicity may be open to the criticism that it colludes with neoliberalism as 'an expression of the cultural logic of late capitalism' (88). There is an important and difficult question being raised here. ANT is certainly open to the criticism that,

although telling us a lot about relational materiality, the network metaphor may also tend to follow too closely rather Machiavellian, managerialist and functionalist strategies of ordering (Law, 'After ANT': 5).

Let us return for a moment to Auran as a site of the emerging creative industries. SAGE is almost the ideal multiple object. Its mode of interpellation, if you will, is *add-on*. It is modular and open, promising through its customizability that it can be remade according to your agenda: use me, link up with me, to make *your* game design and enter the business of game development. What is more, as an end user, you are no longer just a consumer. You too can participate in the process of game development as a producer, a maker of digital content. In their very ontological status, software objects such as SAGE invite us to participate in their making.

In late 1999 Lane again returned to the topic of the importance of SAGE's configurability. He said:

We are exploring the idea of releasing the toolsets that our designers use with the engine to the community; this will assist them with creating the content for mods and add-ons and so on. The extensibility of the engine means that the games we build with it will be very customisable by the end-user. *Dark Reign* did not really support this as fully as we would have liked. It really was too difficult to use and the manuals that we released were too technical.

An extended discussion followed about the relationships between game developers and online gamers. Lane suggested the need to integrate fan content with the core game platform:

At the moment they are kind of separate. You have to leave the game, the game environment, and then use your web browser to surf fan websites and search for new content. You then need to figure out how to integrate that fan-created content with the game. This process can be tricky. But what if we can seamlessly integrate all that right into the game? There is probably a business model here and that needs thinking through. Then there's all of the IP issues around distributing the content to think about. But that's the direction we're thinking in.

Here Lane envisions the close integration of user-created content that we now see with games such as *Spore* and *LittleBigPlanet* some nine years before the release of these titles. In an interview I conducted with Will Wright (designer of *The Sims* and *Spore*) in June 2009, Wright commented that the progression from games such as *Sim City* through to the *Sims* series and *Spore* was fundamentally about the closer integration of user-created

content with the gameplay experience. But the point here is not that Lane is a design visionary. It is more that Lane was thinking through the potential and opportunities afforded by open-architecture game engines and that this generativity is interlinked with questions about the changing relationships between producers and consumers, intellectual property and business models.

Lev Manovich (2001) argues that a defining feature of new media digital objects includes a mode of representation in which we are interpellated as users rather than just viewers or readers (16–17). Manovich tells us that new media function to ‘turn a viewer into an active user’ (183). The digitization of new media objects also enables a reproduction that involves variability and customizability. The same PC hardware technologies and software tools that are used to make and produce these objects are also employed in their consumption and use. This use often involves variation and customizing. Manovich suggests that this is enabled by the modularity of new media objects. He describes this as the ‘fractal structure of new media’ (30)⁹ and suggests that this logic follows or expresses post-industrial logics of ‘just in time’ and ‘production on demand’ modes of production and distribution that value individuality and customization (36; also see 41–2).¹⁰ He refers to computer games as providing a strong example of this trend (120).

Manovich observes that these new media objects indicate a shift in the production process which makes end users feel that they are not just consumers but participants in the process of creating and making a new media object or experience (125). He argues that these objects materialize or reflect this mode of post-industrial capitalism: a ‘set of social and economic practices and conventions is now encoded in the software itself’ (129; also see 131). However, should we follow Manovich’s view that new media objects materialize and reflect the logic of post-industrial or informational capitalism (131)?

Manovich’s understanding of the fractal and modular characteristics of new media and ANT’s approach to technologies both offer important insights into the significance and role of technologies shaping co-creative culture. But perhaps what is missing here is a more precise analytical framework for understanding the mechanisms that coordinate the dynamic relationships associated with technological innovation and change, including how these changes relate to the economic and cultural domains. In *The Nature of Technology*, W. Brian Arthur (2009) offers a theory of technology grounded in the evolutionary and complexity sciences. He sets out from the proposition that technologies evolve and arise through combinations with other technologies. These combinations or bodies of technology include methods, devices, understandings and practices. This is close to the ideas of

assemblage and modularity that I have drawn on throughout this chapter. The difference is that Arthur understands this to be an evolutionary process in the sense that these combinatory arrangements function through an algorithmic evolutionary search mechanism involving processes of adaptation in which new technologies descend from collections that preceded them. He calls this 'combinatorial evolution' through which 'technology creates itself out of itself' (18–23). Arthur contends that if we can understand these evolutionary mechanisms then we can begin to understand the processes of innovation. These combinatory and co-evolving processes are at the heart of dynamics of cultural and economic change, such as co-creative culture.

Arthur suggests that the changes we are seeing associated with digital technologies are precisely about a shift or evolution from fixed industrial-process technologies that characterize the manufacturing economy towards arrangements of technology elements that can be combined and reconfigured for different purposes – a process of 'continual combination' that characterizes genetic engineering and, I would argue, the co-creative media culture assemblages that this book is about. From Arthur's 'combinatorial evolution' perspective then, technologies are not so much independent means of production and more are 'becoming an open language for the creation of structures and functions in the economy' (25). In this framework technologies do not express or reflect a singular 'industrial logic' as such, nor do they determine economic and cultural change. Instead, Arthur argues, as the economy '*encounters*' these technological combinations it reacts by changing and transforming 'its activities, its industries, its organizational arrangements – its structures' (146). As a result of this encounter new processes, new technologies and even new industries emerge.

As these combinations of technologies encounter the economy the resulting commingling and recombining with existing practices and activities can form a new domain of possibilities. Co-creative culture is precisely such a domain. He describes these encounters as a process of 'creative transformation' and 'redomaining' through which industries and cultural practices adapt themselves to the new technologies. This is a co-evolutionary process of 'mutual change and mutual creation' (155). The significance and value of co-creative culture and the associated technologies (in the case of the games industry the engines and editing tools) arise from these co-evolutionary dynamics through which new processes, arrangements, ways of doing and making things emerge not just in one area of application such as the videogames industry but, as Arthur observes, all across the economy.

Arthur's theory of technology and technological change is somewhat close to an ANT approach. The difference and merit of Arthur's work is that

it provides us with an analytical framework grounded in the evolutionary and complexity sciences for understanding these dynamic relationships. In this sense he offers the analytic framework that van Dijck (2013) seeks after noting the analytic limitations of ANT. Rather than just insisting on the relational he offers a rigorous and refined framework for understanding the nature and characteristics of these relational dynamics – they are evolutionary in the context of complex, adaptive systems. I pursue this evolutionary turn in more detail in the final chapter.