EMMC-CSA

European Materials Modelling Council

Report

Expert Meeting on Sustainable economic framework for materials modelling software

July 4, 2018, Cambridge, UK

TABLE OF CONTENT

This workshop and report has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 723867

1. Executive summary

1.1 Description of the deliverable content and objectives

The workshop was held on the 4th of July 2018 in Cambridge. It was attended by 14 delegates including 11 different Software Owners covering a wide range of businesses from start-ups to established ventures. Seven were from outside of the EMMC-CSA, as well as Micress (ACCESS), GRANTA Design, Materials Design and Synopsis/QuantumWise.

Successful software for materials modelling has an expected lifetime of many decades. This long-term nature requires a sound legal and business foundation: the ownership of software must be clearly established and the license models need to be carefully thought through to ensure a sustainable development and maintenance of the software and impactful exploitation by both academic and industrial end-users. Solutions need to be considered for different forms of software including individual software components, comprehensive codes and software platforms. Various business models should be analysed ranging from those built on services around open source software to proprietary solutions.

To archive these objectives, three main topics where covered:

1. Experiences with different business models: to see the status quo
2. Marketplaces: to see how these can aid the software owners
3. Sustainability, maintenance and product life cycle management: to learn about the efforts and find best practice

Webpage: https://emmc.info/events/emmc-csa-expert-meeting-sustainable-economic-framework-for-materials-modelling-software/

1.2 Major outcome

The major outcomes of the workshop are:

- All business models covered in the Discussion Notes were covered and represented at least in some form or as part of the SWO business model.
- Software sales as well as subscription licenses in combination with a range of services (from initial implementation to contract research) are the predominant revenue mix.
- Services play a significant role, with income ranging from 20-80% in many cases. Target software to services ratio is in the range of 70-80 / 30-20. Services are not as scalable but a substantial amount seems required due to the complexity of the software and science. However, there are also exceptions, with some SWO running a successful business with a pure software (and some training etc.) focus.
- SaaS is not widely used though regarded as a potentially way to attract customers who have not the means to get infrastructure and skilled staff in place, and also as a good way to save cost on software installers, installations and implementations. However, industry remains skeptical regarding the security aspects and generally don't accept SaaS.
- For the same reason, the potential for Marketplaces is seen with some skepticism. In addition, the question is whether and to what extent competitors will be willing to use a joint platform (even if basically for marketing) for their software.
- New businesses developing services or SaaS based on proprietary software is somewhat hindered by the lack of a business and licensing model that would be beneficial to both sides. Current licensing is prohibitive for such new businesses.

- Software licensing is overwhelmingly done by means of FlexLM or similar. Trust based licensing is seen as a good idea but difficult to operate and enforce. Dongles have been tried but can cause issues with export delays.
- Sustainability of software requires a change in education and better recognition of the persons in charge.
- Maintenance of software could be achieved in rewriting codes or develop them modular.
- Lifecycle of software requires substantial rethinking and a vision for the future as software's age reaches decades.

2. Progress report (main activities)

2.1 Markets

Materials modelling software serves a wide range of markets including Electronics, Consumer Packaged Goods, Chemicals, Materials, Energy, Manufacturing and Life Sciences.

Market segmentation can be done by considering the position in the value chain where materials modelling is used. The development and engineering sector representing the design **WITH** a given material is far larger than the materials research and design sector where the target is to develop **THE** material itself. There are estimates from market research organisations for the size of the Computer-Aided Engineering market, which roughly represents the former (design WITH the material) but will also include a lot of design software which is strictly outside of materials modelling. The total market size is about $4bn with a CAGR of about 11%. Given that we also know Simulia (part of Dassault Systemes) which represents "modelling with materials" has annual revenue of about $500m, we can assume that the total market for materials modelling software in the downstream sector is surely more than $1bn

Assuming On the other hand, estimates of the market size for materials modelling software in the Design THE material sector is in the range of $100m (a rough estimate based on revenues and numbers of employees in relevant software companies), hence about a factor of 10 smaller than the downstream sector.

About half of the software owners represented at the workshop focus entirely on providing software and services to industry, while the remainder also sell software to the academic research sector. One organisation (Granta) also has a significant market in providing teaching solutions in materials engineering. In terms of industry sectors sold into, all industries including Electronics, Consumer Packaged Goods, Chemicals, Materials, Energy, Manufacturing and Life Sciences are served by the one or the other of the participants. While SWOs provide models that are in principle widely applicable, all have certain sectors they focus on (e.g. semiconductor industry, aerospace industry, chemicals industry)

Some SWOs sell licenses to academia as well as industry while others focus entirely on industry (e.g. in the case of software that is open and free for academics). However, even for those that sell to academia, the share of revenue from academia is declining as commercial sector income grows faster.

2.2 Experiences with different business models

The following summarises experiences shared regarding different business models. Almost all participants have business models based either entirely or largely on proprietary software, rather on open source codes. The exception was one organisation considering starting up a business based on consulting services around an open source code. Nevertheless, open and or free components play a role in the mix of almost all software owners. There does not seem to be a dominant pattern of mixing proprietary and open/free software however. In particular, some SWOs offer proprietary pre- and post-processing tools (including GUIs) around open source codes while others do the opposite (i.e. offer a free GUI or integration platform for proprietary software). The

latter applies in particular where the overall solution for the customer depends not just on one model, but a workflow integrating a range of models.
Also, almost all SWOs run hybrid business models, in particular a combination of software and services. However, it was also pointed out that it can become complex to manage different business models within the same company and some business schools advise against it.

2.2.1 Product Sales and maintenance services

This is used by the majority of software owners (70% of those at the workshop). The product typically comprises a whole package consisting of the simulation engine(s) including pre- and post-processors, interfaces, specialised GUIs, data and databases, all or most of which have proprietary licensing. Open source components are sometimes included, where relevant. Mostly, open source components play a role in underlying libraries are shipped with the product (e.g. Python).
Product Sales means that the license to use the product is perpetual, but typically maintenance (including support for new or upgraded operating systems and libraries) is included only for one year. The revenue is therefore derived from new license sales (i.e. more licenses at current customers, new products at current customers and new customers) as well as maintenance. The maintenance charge is in the region of 15-20%. Most SWO also sell additional services (see below), e.g. comprising training, consultation and implementation.
There are different price points for customers in commercial organisations and academia. The latter is further differentiated by pure research and allowing commercial consultancy.

2.2.2 Subscription based licensing

Subscription based licensing provides access to installed software for a period of time, typically one year. The annual subscription fee includes maintenance. It is also a widely used revenue type, adopted by 60% of the Software Owners at the workshop. Note also the wider discussion in the industry and the general trend in the last years to move from perpetual to subscription licenses.[1]

2.2.3 Consortium model and Public Funding

Customers Consortia are regarded as very important by several SWOs. The idea is to let the customers define the design requirements for the software, define priorities, test, and provide comments and feedback. Consortia support building and maintain a community and push and pay for development of new functionality. Three of the ten Software owners run customer consortia. None has a consortium with other software owners. The SWO involved in these consortia select the consortium members very carefully and give them preferential access to software for a certain time. The value for a consortium customer is that they get the software earlier and can take advantage before it is broadly released.

2.2.4 Services and consulting (based on proprietary or open source software)

The overwhelming majority of software owners offer services, in the case of the workshop participants it was 7/10. These range from very limited service offerings that are often included in the license fee to implementation, customisation and contract research services. Maturity of the software was also seen as a key factor, with start-

[1] "MSC Software recently introduced MSC One, a system that lets customers buy subscription tokens that can be applied to just about any product in MSC's portfolio. Before (and still, if that's the buyer's choice), one bought perpetual licenses for a specific product. With MSC One, buyers gain access to MSC's newer products and acquisitions." https://www.linkedin.com/pulse/msc-hexagon-where-simulation-gets-real-monica-schnitger/

ups and less mature solutions requiring a higher percentage of services. Revenue from services makes hence a substantial part of income for materials modelling software.

Software vs Service
100% software, no services other than maintenance
30% : 70%
0%:100% to start with 80%: 20% to end with
50%:50%
60%:40%
80%:20%

It became apparent that there is no Gold-standard with respect to how many % of services should be offered – it depends very much on the product at hand.
Reasons for offering services and consulting include:

- Implementation services are very important to make sure the software fits to whatever workflows the company requires.
- The software is very complex and needs consultancy. They are setting up simulations and they charge for these services. They idea is then that the customer can learn how to adapt the software but rarely can learn how to setup from scratch.
- The huge variety of applications of materials modelling means that the customer may profit from purchasing services that can aid with mastering the software faster to get results.
- Software complexity and maturity (or lack of).
- In depth subject knowledge in the field of the software. The main value in the case of open source software-based services comes from people with knowledge/expertise

Caveats regarding services are that (a) they don't scale in the same way as software in terms of revenue and profitability (one can sell people per the hour/day only once, whereas software can be sold several times) and (b) service engagements go through peaks and troughs, which can lead to resourcing issues. One SWO addresses the issue by working with universities that are "centres of excellence" for their software and can provide some services.

2.2.5 Business models based on open-source and free to use software

"Open source software is software with source code that anyone can inspect, modify, and enhance."
Cited from https://opensource.com/resources/what-open-source
While Open Source does not necessarily mean 'free of charge', we focus here on software/codes that do not have a license requiring charging.
At the workshop, an open-source based business model was represented by an academic group considering a start-up on that basis. The main value comes from people with knowledge/expertise and this is their asset. The aim is to offer consultancy and solve customers' problems. Also, they look into cloud-based SaaS and charge customers per hour/subscription.
Training and support may be charged for as well. The may also want to do software enhancements for a particular customer, and the outcome would then benefit only those.
While none of the established companies represented at the workshop had a business model based mainly on open source software, four companies distribute software that includes open source codes. In addition, some

companies focus entirely on the commercial software market, while free and/or open source versions of the code are available separately to academics for research purposes.
Reasons for choosing open source include that its visibility makes collaboration easier and enables easier feedback and bug reporting. It is also a "nothing to hide" approach as everybody can look anytime into every line of code. Also, it enables a SWO to have a stake/share in a community around an open source product. It can give the SWO credibility and visibility and could be a sweet spot for open source-based business models.
Revenue is based not only on services, but also on OS-supported and ready-to-install versions, pre- and post-processing tools as well as materials relations (e.g. forcefields) that work with the models encoded in open source software.
A special case is a SWO that develops the software to support in-house projects. Due to an increase in external collaborations, some codes were made open source to make collaboration easier and increase visibility.

2.2.6 Government funding

Contributions from government funded projects are important to most software owners, with 7/10 taking part on such projects with funding from national, EU and in some cases US sources. In particular, new developments as well as standardisation and sustainability of software happen often via special funding from the EU or international excellence programmes.

2.2.7 SaaS

Currently, SaaS is not widely used since industry remains skeptical regarding the security aspects and generally doesn't accept SaaS for business-critical applications and data. Companies prefer the software to be installed behind their firewall rather than be cloud hosted. That holds even in cases that the software is web-based and hence could potentially be provided as a hosted service.
One start-up is aiming to use SaaS with a usage-based payment model. looking into the best possible license model – e.g. a node-based license model may not be too good for them.
partnering with a software provider rather than buy as they have no permanent usage for software. In return, they may offer new business models for the SWO and to find new opportunities.
However, SaaS is regarded by SWOs as having the potential to attract customers that do not have the means to get infrastructure and skilled staff in place. For example, it could lead to getting into market niches, especially SMEs, who could not afford a code and the infra structure needed (hardware, skilled personnel, financial investment to get started). SaaS is also as a good way to save cost on software installers, installations and implementations. Also, it is seen as a good way to utilise substantial knowledge around a particular opensource software and to sell simulation services to experimentalists, commanding a premium for best practises/correct workflows.
A barrier for new businesses to emerge it that they need to purchase/license software from a SWO to offer a specific Software as a Service (SaaS). The start-up costs are can be prohibitive, however, as the licensing is expensive.

2.2.8 Data

In addition to the meeting host (Granta) that of course sell licenses to materials data, some other SWO also sell data that support the modelling, e.g. databases of structures, or thermodynamic data. Also, users are supported by enabling direct connections between models and relevant databases.
In addition, one organisation is able to offer validation plus experimental data as well.

2.3 Stakeholder/Customer management and value generation

Besides the Business Model, skillful stakeholder/customer management and value generation also play a substantial role in the day-to-day life of an SWO.

Some of the SWOs try to provide a full/holistic solution for their customers and try the utmost to keep them hooked in their environment. This goes hand-in-hand with regular reviews to see what their customers are interested in. This happens often via user group meeting, attending fairs, and training events – world-wide. Annual user group meetings are widely used as a means to stay in contact with users.

Some software owners come from academia and cater for this audience by providing teaching materials and a lower priced licensing scheme. In some cases, where software was developed as part of national funding, the software has to be provided for free to academics and government institutions. There is also an interest in involving academia to lead a new generation to use the software product.

The SWO is often confronted with the fact that customers may have already a plethora of codes and that their product is used additionally. Thus, many SWOs allow implementation to other codes and if they have GUIs, they make them interoperable, i.e. outputs from competitor/open source codes can be read or they permit generation of input files. Some SWOs would also interface to popular opensource codes. Customers also prefer to get software in one transaction, install it and be ready to run.

2.4 Start-up models and experiences

Participants shared experiences with starting up and scaling up a software company. Examples included:

- Establishing a firm base in a specific market segment (such as teaching software and courses in the case of Granta) and branching into other segments (industry, starting with specific industry sectors) etc.
- Spin-off with Venture Capital funding: Proprietary software. Caveat that it can go wrong. Organic growth with solid proprietary technology base works better (and led to take over by global player).
- Starting up as vendor of an established and academically widely used code to sell to industry.
- Start-up as a consultancy with high value know-how but relatively immature code. Derive income from consulting to hire software engineers and change from "code" to professional software. Transition to robust software while reducing the consulting/services element.
- Software sales within not-for-profit organisation, i.e. keeping some sort of hybrid between research sector and software sector.
- Consultancy based on open source software, potentially combined with SaaS that solves the issue of installation and configuration
- Very targeted workflows and services, in particular as SaaS for experimentalists. Issues include the need to negotiate license agreements with any proprietary codes that are required for the solution. Currently terms tend to be prohibitive.
- Proprietary pre- and post-processing solutions (GUIs, integration etc) around open source software, funded by investment and organic growth.

2.5 Marketplaces

The question is whether and how the emerging EU market places can support business and distribution for SWOs.

Existing marketplaces were briefly covered in the introductory talk, including NanoHub (well known in the materials modelling community), Fortissimo (still run as an EU project), 3DExperience marketplace for additive

manufacturing and ScienceExchange[2]. Interesting about the 3DS marketplace is the business model not being based on increased software license revenue[3]: "You go to the marketplace and you work on Catia in your browser without the need for a license (otherwise €15,000). You pay for the result of this work under Catia and the connection with manufacturers or other stakeholders."
The SWOs did not entirely warm up to the idea of marketplaces but were not strictly opposed either.
There were some discussions about barriers and potential solutions. One of the key issues will be how to reach critical mass of users.

Barriers

- EU funds are only for a few years, so there is a worry what will happen afterwards. It takes a long time to build up a critical mass of users, as the NanoHub example showed. Contrast that with Nanohub (US) receiving 10s of millions of USD funding over more than a decade now.
- Potential customers are skeptical to put data on some site they see as not trustworthy/big/anonymous. They prefer a trusted vendor/SWO.
- Close contact with SWOs is regarded as essential in the materials modelling field in order for the users to derive full value from the code. Marketplace may make the customer interaction more remote, i.e. once further removed than it is today.
- Big companies have their own market places and set their own 'standards'.
- The cloud and running calculations outside of a firewall.
- Market segment in industry that might use marketplaces is regarded as small.

Solutions

- Enable closer contact between SWO and customer.
- VC funding may be available to marketplaces, see ScienceExchange, that has so far received $60m.[4]
- Enable consultants/translators with a good reputation to bridge between the customer and an anonymous market place.
- Facilitate secure interaction and enable strong feedback between users/customers and software owners.
- Integration plays a big role and new Marketplaces (Marketplace, Vimmp) will cover this.
- Let them be an infrastructure for SaaS and an open, free accessible, well documented exchange structure that anybody can plug into without having the barrier to buy a specific product.
- Find ways to run calculations inside the firewall of a company.
- Focus of marketplaces initially on early adopter market sectors and related models.

Opportunities

- For new codes and startups, the marketplace could serve as an intermediate staging place until smaller players become more widely known.

[2] ScienceExchange is mainly focussed on facilitating Contract Research services, mostly in the Life Sciences industry, and adds value by providing a secure platform for outsourced research, providing the ability to efficiently access services from qualified providers, all with pre-established contracts in place that protect client intellectual property and confidentiality.

[3] https://www.usinenouvelle.com/article/l-acces-a-catia-solidworks-et-tous-les-autres-logiciels-de-dassault-systemes-bientot-gratuit-et-en-ligne.N524554

[4] https://www.forbes.com/sites/alexkonrad/2017/06/29/science-exchange-takes-on-28m-to-help-companies-outsource-rd/

- They can be seen as infrastructure for SaaS and an open, free accessible, well documented exchange structure that anybody can plug into without having the barrier to buy a specific product.

2.6 Sustainability, maintenance and product life cycle management

2.6.1 From academic code to professional software

In almost all cases, materials modelling software has its origins in academia. However, the quality of the code is often poor. It was often coded by PhD students or PostDocs and these tend to focus on their next publication. The long-term maintainability is less of their concern and they are not typically trained in software engineering. Research grants also focus on exciting new science and do not tend to reward writing clean code. Likewise, grant proposal reviewers are typically not concerned with software quality and sustainability, even if it is made a criterion by the funding agency.
It hence takes considerable effort on behalf of commercial software owners to upgrade or rewrite code, and some examples were represented which took years to complete.
Open source plays of course a big role in academic software developments. A wide range of points were discussed in this regard, including:

- Open source makes distribution easy and facilitates collaboration and wide-spread uptake.
- Open source codes can train people on developing codes.
- Open source development by a community needs strict coordination of contributors. If software developer let other people enhance their code they may spend up to 40% of their time just doing that.
- Code quality is a problem as open source easily diverges.
- "Leaders" that can keep a community together and 'gate-keepers' for quality are important.
- Open source can push the quality of commercial codes.
- Commercial software owners contribute important testing, bug fixes and are often the only ones that know how to compile/install under Windows and hence get community requests for that: relationship can be mutually beneficial.

2.6.2 Build, buy and partner

Some software owners mainly develop their own code but also may consider paying royalties if a certain product was suitable to enhance their offerings. They may not write their own code, if the 3rd party code is already well used by the community and the final income due to having a better offering may outweigh the royalty costs. Often a commercial SWO may act as the sole reseller of an academic code.
When coding, some developers will work with libraries which are needed to maintain and develop the software. These come with a licensing of their own and the SWO has to make sure all is in line with the licensing of their actual product.
Often databases play a role in materials modelling. These are typically added based on partnerships and royalty-based distribution agreement. The latter is mostly relevant to industry customers as academia may have free access to certain databases anyway.

2.6.3 Software license management

Some license management solution such as FLEXlm[5] or LM-X[6] were discussed, with FLEXlm being the most widely used. The type of permissions and limitations vary and need to be carefully considered. Typically, multi-CPU licenses have become the norm, with limitations on the number of simultaneous jobs.
FLEXlm can be used as back-office as it contains user management tools. This could be used for statistics, to see what feature of software has been used for how long, and by whom. These logs can be analysed and any use can be traced.[7]
"Paper licenses" (trust-based licensing) seemed not to be used by any SWO present, however the principle is seen positively. One can give a company such a license, ask them after a year how many users there were, and then adapt the software offer.
Dongles seemed to be rather disliked. They need white listing for a given machine at a given place. One SWO did sell a dongle, and it had to be white listed by the buyers IT. It was too much paperwork to resolve. Customs was also found to be tricky as they check everything that looks like a USB stick.[8]

2.6.4 Sustainability

Sustainability of a code is vital to generate income for commercial codes, but it is also vital to guarantee survival of an open source code. It was argued that there should be more funding for best practises and software quality. There is a lack in training best practices in software development and HPC. There are still not many university type courses on offer. Even if students take these courses, they can be disappointed because they wanted to do science rather than developing code.
Materials modelling codes tend to have a much longer lifetime than typically was originally planned when they were first written. This long-term planning needs to be put into academic developers' minds.
In the UK there is the Software Sustainability Institute looking after sustainable software. The UK also has an association of Research Software Engineers. On their webpage they state: "*We are an association working to create a community and raise awareness of the UK's Research Software Engineers. We campaign for the recognition and adoption of the RSE role within academia along with the need for appropriate reward and career opportunities for RSEs."* There is also ARCHER, a national Supercomputer Centre in Edinburgh, that provides training in best practices for software developers.

[5] "FLEXlm software is a prominent license management solution that enables software vendors to impose restrictions on the number of software seats available to their customers. FLEXlm supports different licensing policies such as Floating (aka Concurrent) and Node Locked licenses. This type of software system is also referred to as DRM (Digital Rights Management) Solutions."

From: https://www.openlm.com/what-is-flexlm-what-is-flexnet-2/

[6] "LM-X License Manager allows software vendors to license their software products to end users in a secure way. LM-X consists of several libraries and tools that can be used by software application developers to protect products against potential overuse, control their license policies externally from applications and enforce various levels of security."
From: https://www.x-formation.com/lm-x-license-manager/

[7] Note however that there may be restrictions due to company rules as well as privacy legislation.

[8] Note however, that another SWO not present at the meeting has customers that are happy using Dongles since they the issue of having to connect to a license server.

2.6.5 Maintenance and Life Cycle Management

Software maintenance includes continuous testing during new developments and following each compilation, bug fixes, updates/upgrades of libraries, maintaining current and older OS support while adding new Operating Systems, testing and updating installers, updating documentation and training, etc. All the issues mentioned above are relevant and if not handled well may consume a vast amount of resources. For example, at one SWO typically one person works on just upgrading libraries. Hence controlling the amount of maintenance is an important topic for SWO. See also the discussion about SaaS which would reduce the costs considerably.
Also, there is a lack of standardisation of interfaces: SWO need to maintain to many of these. Despite efforts to agree on common formats etc., it is seen as unlikely that big players in particular will diverge from their proprietary solutions.
Software release cycles are typically annual but for less mature ones they can be 2-3 times per year. Customers don't seem to like maintenance cycles < 1 year as they often have efforts to reinstall new software.
Code tends to grow so at some point it cannot be easily extended anymore. Adding new features and cleaning the code should go hand in hand - maintenance and developing new features become closely fused together. The more dependencies one adds to their software the more care and planning is need for what is needed over the lifecycle. Software often becomes so complex that it has to be rewritten at some point. This is a big challenge in terms of cost.
Customers sometimes request information on tools they used several years ago. Some SWO hence maintain archives including the code and a complied version.

3. Conclusions

The major outcomes of the workshop are:

- All business models covered in the Discussion Notes were covered and represented at least in some form or as part of the SWO business model.
- Software sales as well as subscription licenses in combination with a range of services (from initial implementation to contract research) are the predominant revenue mix.
- Services play a significant role, with income ranging from 20-80% in many cases. Target software to services ratio is in the range of 70-80 / 30-20. Services are not as scalable but a substantial amount seems required due to the complexity of the software and science. However, there are also exceptions, with some SWO running a successful business with a pure software (and some training etc) focus.
- SaaS is not widely used though regarded as a potentially way to attract customers who have not the means to get infrastructure and skilled staff in place, and also as a good way to save cost on software installers, installations and implementations. However, industry remains sceptical regarding the security aspects and generally don't accept SaaS.
- For the same reason, the potential for Marketplaces is seen with some scepticism. In addition, the question is whether and to what extent competitors will be willing to use a joint platform (even if basically for marketing) for their software.
- New businesses developing services or SaaS based on proprietary software is somewhat hindered by the lack of a business and licensing model that would be beneficial to both sides. Current licensing is prohibitive for such new businesses.
- Software licensing is overwhelmingly done by means of FlexLM or similar. Trust based licensing is seen as a good idea but difficult to operate and enforce. Dongles have been tried but can cause issues with export delays.

- Sustainability of software requires a change in education and better recognition of the persons in charge.
- Maintenance of software could be achieved in rewriting codes or develop them modular.
- Lifecycle of software requires substantial rethinking and a vision for the future as software's age reaches decades.

4. References

See in text.

5. Annex1: Revenue Models and Software vs Services polls

The participants were asked to post the revenue models they use, with the following lettering/numbering:

A. Product sales and maintenance services
B. Subscription based licensing
C. Consortium model
    1. Software industry consortium incl joint ventures
    2. Customer consortium
D. Services and consulting based on proprietary software solutions
E. Open-source based business models
    1. Consulting, based on an open source product.
    2. Loss-leader/market positioner: e.g., an open source client
    3. Proprietary product with a license that guarantees open source release after a certain time
    4. Certified version of an open source product (e.g., RedHat).
F. Government funding
    1. assistance awards and contracts
    2. federal data facility support etc.
G. Software as a Service (SaaS)
    1. Capacity-based Model
    2. Feature-based Model
    3. Time-based Model
    4. Use-case Model
    5. Quality of Service Model

RESULTS

A, B, D (implementation services), C2, F1
A-C2-D-E1-F1
A, D, E1
D, G1, F1
G1, G2, G3, G4
A, B, D, F, G
A, B, C2, D, indirectly F
B, to a lesser extend A, F
B, D, E2
Now: A/B, F, (E2), Future: A/B, F, G

Also, the participants were asked for their current and/or proposed revenue share of services versus software:

Software vs Service
80% Software, 20% EU funding, so basically 100% software
30% : 70%
0%:100% to start with 80%: 20% to end with
50%:50%
60%:40%
60%:40% (20% maintenance,20% services, so basically 80:20

6. Annex 2: Agenda

6.1 Scope

Successful software for materials modelling has an expected lifetime of many decades. This long-term nature requires a sound legal and business foundation: the ownership of software must be clearly established and the license models need to be carefully thought through to ensure a sustainable development and maintenance of the software and impactful exploitation by both academic and industrial end-users. Solutions need to be considered for different forms of software including individual software components, comprehensive codes and software platforms. Various business models should be analysed ranging from those built on services around open source software to proprietary solutions.

6.2 Meeting Format

Given the potentially sensitive nature, the so-called Chatham House Rules [1] will apply.
Two sessions are planned:

1. Business models and value creation
2. Sustainable software development, maintenance and life cycle management

Each session will include a number of short presentations from SWOs, planned so that different types of business model etc. will be covered and discussed.
Aim of the discussion is to further probe pros and cons and make recommendations about good practice for each type. The aim is not to give any value judgement but rather to provide more insights and recommendations that will help to support future sustainable software deployment, speeding up the transition from academia to industry, ensuring long term sustainability and growth of the software sector.

6.3 Agenda

09:30 – 10:00 Arrival/Coffee
10:00 – 10:20 Welcome, Introductions, Ground Rules
10:20 – 10:40 The market for materials modelling software and services: Overview (G. Goldbeck)
10:40 – 11:00 Discussion on markets and trends
11:00 – 11:50 Impulse talks on: Experiences with different business models, including

- License/distribution type
  - Proprietary, owned by company
  - Closed source, distributed by company
  - Open Source, developed by company
  - Open Source, not developed but distributed by company
- Software type:
  - Individual codes/components
  - User Interfaces
  - Integrated environments
  - Platforms
- Revenue types
  - Software license sales
  - Maintenance
  - Software license subscriptions
  - Software as a Service
  - Services: Software installation/implementation/customization

Address the question of where value is created (and hence people pay)

11:50 – 12:40 Discussion on business models and value creation
12:40 – 13:30 Lunch
13:30 – 14:20 Impulse talks on: Experience with sustainable development, maintenance and life cycle management of software
14:20 – 15:20 Discussion on sustainability, maintenance and product life cycle management
15:20 – 15:40 Break
15:40 – 16:20 Formulating summary and recommendations
16:20 – 16:30 Conclusions and Close of meeting

7. Annex 3: Discussion Notes

7.1 Topic of the Workshop

Successful software for materials modelling has an expected lifetime of many decades. This long-term nature requires a sound legal and business foundation: the ownership of software must be clearly established and the license models need to be carefully thought through to ensure a sustainable development and maintenance of the software and impactful exploitation by both academic and industrial end-users. Solutions need to be considered for different forms of software including individual software components, comprehensive codes and

software platforms. Various business models should be analysed ranging from those built on services around open source software to proprietary solutions.

7.2 The objective of this meeting

The aim of this meeting is to provide insights and recommendations that will help to support future sustainable software deployment, speeding up the transition from academia to industry, ensuring long term sustainability and growth of the software sector.

7.3 Discussion points and questions

7.3.1 Discussion Point 1: Business models and value creation

1. What is the best Business Modell for material modelling software
   - More and regular Innovation?
   - Creating "blue oceans" of uncontested market space, as opposed to "red oceans" where competitors fight for dominance (W.C.; Mauborgne, R. (2004). Blue Ocean Strategy: How to Create Uncontested Market Space and Make the Competition Irrelevant. Boston: Harvard Business School Press)?
   - Software as a Service (SaaS)?
   - Do you use open source software to create or maintain a market position for proprietary software? (e.g., an open source client or free GUI)
   - Do you publish open source drivers for proprietary hardware, both for peer review benefits and also to allow operating system vendors/maintainers to adapt the driver to future changes in system interfaces?
   - Do you use expertise in an open source product to drive revenue for packaging and/or consulting services?
   - Do you sell books or other accessories to open source products (e.g., O'Reilly publishers)?
   - Do you sell a proprietary product with a license that guarantees open source release/"perpetual license" after a certain time, in order to guarantee future maintainability to prospective customers?
   - Do you charge for the branded, trademarked, tested, and certified version of an open source product?
   - What comments and views could you share about sustainable business models for materials modelling software?
2. E. Raymond points out ("The Magic Cauldron," The Cathedral and the Bazaar"; Sebastopol, CA: O'Reilly, 1999) that in most cases, open source software does not have as much revenue available to fund its development as is provided by license fees of proprietary software.
   - Do you agree, that proprietary software often leads the development of end-user software?
   - Do you see that open source provides low-end users with an inexpensive alternative, and also provides an open and flexible platform for those who need custom modifications?
   - Do have the impression that proprietary products must therefore keep innovating to push the frontier forward, as its open source competitors catch up behind them?
3. There is an agreement that users purchase software because it is of some sort of value to them
   - When using your software, what value does it create for the user?
   - Why do users buy your software?
   - Why don't users buy your software?
   - How will the user capture this value?
   - What is the biggest barrier for a user not to user your software?

7.3.2 Discussion Point 2: Sustainable software development, maintenance and life cycle management

1. **Software should be developed sustainable, but is this easy to achieve?**

- What are your resources requirements to create and maintain your software professionally?
- What costs do you have in doing the above, i.e. the percentage of your profit that will feed into the above?

2. **Maintenance**

- Do you charge maintenance?
- How significant is the income for software maintenance?

3. **Every software product needs Life Cycle Management.**

- How many years do you keep/support legacy products?
- Looking at 100% of your developer force how many % work: (On legacy, Operating IT, Maintenance/bugs, new development?)

4. **What needs to happen to bring software out of academia to the market?**

- Better quality of coding?
- Support by better national and international funding agencies?
- University investments?
- Better consultancy for spin-offs at universities?
- Marketplaces?

7.4 Background

7.4.1 Materials modelling software market

Relevant studies include the following two studies attempting to do this, are:

- The Scientific Software Industry[i], addressing the following topics:
  - The structure of the software industry.
  - Requirements for software development: in-house and through collaboration.
  - Routes to market for scientific software, e.g. via software houses or direct licensing into specific industries.
  - Commercialization requirements: standards, IP ownership, licensing schemes.
  - Warranty and liability issues.
- Schnitger Corp reports, https://www.linkedin.com/in/monicaschnitger/

Computer-Aided Engineering market:

- https://www.transparencymarketresearch.com/computer-aided-engineering-market.html
- https://assetsstock.com/technology/global-cae-software-market-vendors-dassault.html
- https://www.businesswire.com/news/home/20171020005370/en/Global-CAE-Market-2017-2021-Growth-Application-Specific-CAE
- https://www.zionmarketresearch.com/news/computer-aided-engineering-market

The CAE market currently generates global revenues of 1.2 billion €. Also, it is growing at an annual rate of 11% (according to TechNavio's report). The global CAE market is forecast to grow at a CAGR of 11.84% during the period 2017-2021.

Zion Research has published a report titled "Computer Aided Engineering Market (Finite Element Analysis and Computational Fluid Dynamics) for Aerospace, Automobile, Electronic and Electricals, Defence, Industrial Machineries and Other Applications called "Global Industry Perspective, Comprehensive Analysis and Forecast, 2015-2021". According to the report, the global demand for computer aided engineering market was valued at around USD 3,062.37 billion in 2015, and is expected to reach approximately USD 5,863.36 million in 2021, growing at a CAGR of slightly above 11.10% between 2016 and 2021.

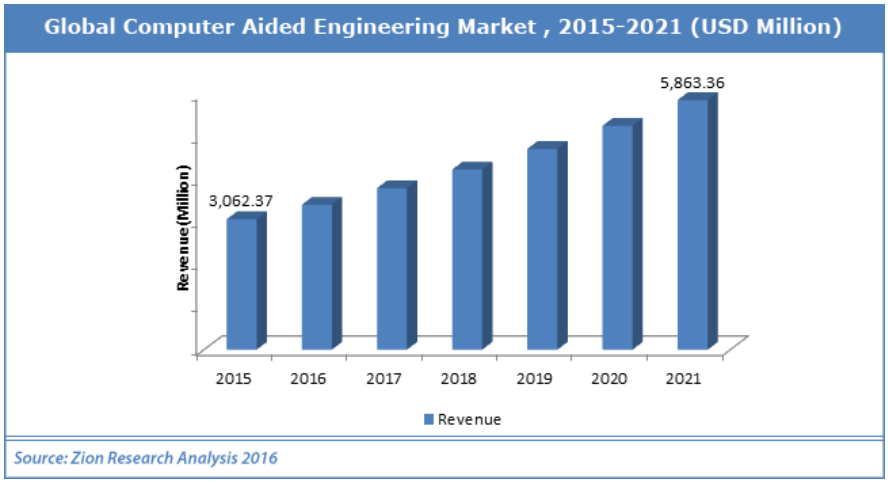

Figure1: Computer Aided Engineering Market

7.4.2 Software sustainability: general considerations

According to a report from the Sustainable Business Models Team to the Brokering Governance Working Group of the Research Data Alliance (RDA) [ii], software sustainability is influenced by a range of factors as shown in Figure 1.

Figure 1: Required attributes for software sustainability adapted from Ref **Fehler! Textmarke nicht definiert.** and the Software Sustainability Institute's definitions for software sustainability (http://www.software.ac.uk/).

7.4.3 Business Models

7.4.3.1 Overview

Software services for research provide intangible goods and services, and frequently employ business models different from those utilized by companies providing goods and services focused on financial, physical or human sectors. As they are currently configured, those entities responsible for software serve or could serve four classic business archetype roles.

- A creator that transforms ideas into a product
- A distributor of software
- A lessor that provides the rights to use the software

An underlying premise of the archetypes with respect to software is that the provenance and intellectual property (IP) are established.

Software businesses may use a hybrid approach serving the role of more than one archetype function since they are acting as both the inventor and the IP lessor.

The following compilation and analysis of business models in this report is based on the following

- The seminal essay by Eric Raymond "The Magic Cauldron"[iii]
- Report from the Sustainable Business Models Team to the Brokering Governance Working Group of the Research Data Alliance (RDA)[ii]
- RDA discussion paper on Software as a Service business model[iv]
- Publications on marketplaces[v, vi, vii]

Business Models relevant to materials modelling software include the following:

- Product Sales and maintenance services
- Subscription based licensing
- Consortium model
- Services and consulting based on proprietary software solutions
- Open-source based business models
    - Consulting, also known as "give away the recipe, open a restaurant:" use expertise in an open source product to drive revenue for packaging and/or consulting services (e.g. OpenFOAM).
    - Loss-leader/market positioner: use open source software to create or maintain a market position for proprietary software (e.g., an open source client creates a market for a proprietary server) and to a lesser extent "Accessorizing" (sell accessories to open source products)
    - Free the future, sell the present: sell a proprietary product with a license that guarantees open source release after a certain time, in order to guarantee future maintainability to prospective customers (e.g., Alladdin GhostScript).
    - Free the software, sell the brand: charge for the branded, trademarked, tested, and certified version of an open source product (e.g., RedHat).
- Government funding (assistance awards and contracts, federal data facility support etc)
- Software as a Service (SaaS)
- Marketplace business models [v] and strategies [vii]
    - Product community
    - Offline services on-demand
    - Online services marketplace strategies
    - Defensive one-stop shop

- Distribution channel extension
- Business model transformation

7.4.3.2 Open Source based business models

According to Raymond, there are the following business models

- Loss-leader/market positioner: use open source software to create or maintain a market position for proprietary software (e.g., an open source client creates a market for a proprietary server).
- Widget frosting: publish open source drivers for proprietary hardware, both for peer review benefits and also to allow operating system vendors/maintainers to adapt the driver to future changes in system interfaces.
- Consulting, also known as "give away the recipe, open a restaurant:" use expertise in an open source product to drive revenue for packaging and/or consulting services (e.g. OpenFOAM).
- Accessorizing: sell books or other accessories to open source products (e.g., O'Reilly publishers).
- Free the future, sell the present: sell a proprietary product with a license that guarantees open source release after a certain time, in order to guarantee future maintainability to prospective customers (e.g., Alladdin GhostScript).
- Free the software, sell the brand: charge for the branded, trademarked, tested, and certified version of an open source product (e.g., RedHat).

Raymond's essay also pointed out that in most cases, open source software does not have as much revenue available to fund its development as is provided by license fees of proprietary software. For this reason, proprietary software often leads the development of end-user software, while open source provides low-end users with an inexpensive alternative, and also provides an open and flexible platform for those who need custom modifications. Proprietary products must therefore keep innovating to push the frontier forward, as its open source competitors catch up behind them.

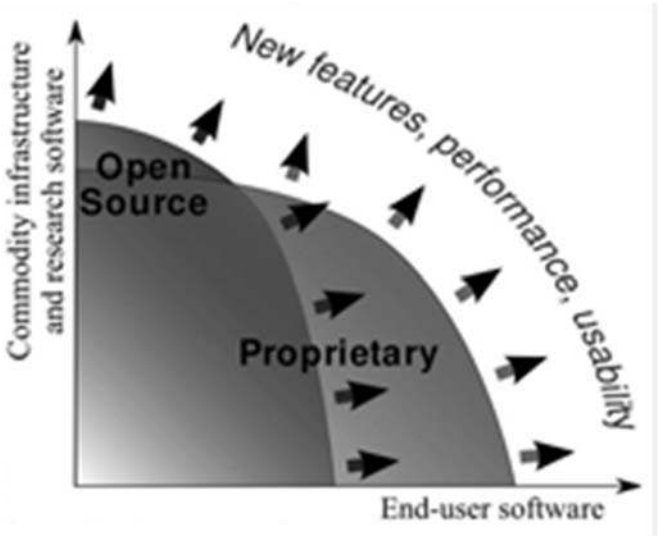

Figure 3: Eric Raymond's concept of expansion of open source and proprietary software capabilities.

7.4.3.3 Software business models (RDA)

The RDA report discusses five classes of business models. For each, the strength and weaknesses in the context of long-term sustainability is discussed. For their type of software (brokering middleware) they find that a hybridized model incorporating aspects of three different business models over the lifespan, i.e. federally funded data facility guardianship in the establishment stage replaced or supported by a Consortium model and/or Software-as-a-Service as the software matures, will likely provide the strongest model for sustainment.

Software companies can differentiate their business model by offering software as a product or, software as a service (SaaS), or a combination of both. As K.M Popp (2015) notes, SaaS means the software vendor does not deliver the software, but the customer gets both access to the software and usage rights […] The software vendor carries the cost of software support, maintenance, and operation."

7.4.3.4 Revenue Models

The software industry has developed a number of business models to generate revenue to support all or some of the attributes of sustainment outlined above. SaaS is considered in this review of viable revenue models, as well as non-commercial and hybrid funding models. The revenue models explored are:

- Government funding through assistance awards and contracts
- Government funding through federal data facility guardianship
- SaaS (including tiered pricing);
- Information and Ad sales
- Corporate Support (including Foundation Support) and Product /Service Sales, (foundation support)
- Consortium model.

Each of these revenue options are discussed in the following context and may have variants depending on the target market:

1) Description of the business model with examples of software currently supported under that model
2) Mapping Revenue Models onto Sustainment Attributes
3) Characteristics of the business model that would benefit the sustainability of software
4) Challenges in the application of the model to support software for the research and education communities

7.4.3.5 Government Funding Through Assistance Awards and Contracts

Model Description

For decades, governments have provided funding to support the development and sustainment of software. Grants are assistance awards that, in the context of the interests of the Research Data Alliance, are often used to "assist" university and non-profits in the pursuit of a research problem. The supporting government entities generally have little involvement in guiding the research activity that is funded by grants.

Globus, NetCDF, and the Community Climate Model are examples of software that have been developed with government support and continue to be refreshed and modified with U.S. Federal dollars. The longevity of support for these software packages is measured in decades.

This sort of funding, for example, has significantly enabled Australian researchers to participate in eResearch by supporting the development of IT infrastructure, software, tools and services for open publication, discovery, sharing and use of Australia's research data outputs through the Australian National Data Service (ANDS) and National eResearch Collaboration, Tools and Resources (NeCTAR).

Mapping Revenue Models onto Sustainment Attributes

Funding – Requires a proposal. The home institution of the principal investigator bears part or all of the cost of preparing the proposal. Government initiated strategic investments, can be of longer duration than research grants and cooperative research agreements, but are nonetheless limited in their duration.

Users – The concept of market development is interpreted to mean understanding which research communities might find the software useful in the pursuit of their objectives. This is an important observation that could have an impact on sustainability. The design of the software could be influenced by the applicability of the software to other venues. Government funding does not necessarily encourage or discourage the principle investigators from engaging in market development.

Communities –Community engagement is often encouraged and sometimes additional funding for outreach and engagement activities may be provided.

Human Resources –Most often the award is made to the institution which employs the researcher. Often this institution provides tangible and intangible contributions. Under these circumstances software developed under a research project 1) has the revenue stream for the work virtually assured for a limited period of time, 2) can be flexible in design and execution, 3) has a natural partner in the institutional home that often provides benefits useful to the project, and 4) has the potential to contribute to the educational mandate of the institution.

Software Engineering – Investing in good software engineering is an important use of the revenue obtained from the Government. However, government funding of a project, particularly in a university setting, does not assure that good software engineering practices will be used throughout the project.

Product Management – Under government funding, product management is usually the responsibility of the institution that received the funding and the principle investigator employed by the institution. In general, the government usually allows the Intellectual Property rights to reside with the institution where the software was developed but this may require the government to have royalty free access to the software.

Advantages

Advances the conceptualization, prototyping, and initial testing of the software for such attributes as functionality and robustness. There are some instances of longer-term support of software that the funding agencies deem necessary for significant community functions such as weather prediction.

Challenges

- Funding term limits do not create sustainment.
- Government initiated strategic investments are often politically motivated and at the mercy of changes in Government leadership and party politics.
- As noted earlier, government support for software development does not ensure that good software engineering practices are employed.
- When software is created for one purpose and used by its developer, ease of operation by others or an outside organization is not generally a concern of the developer.

7.4.3.6 Government Funding through Federal Data Facility Guardianship

Model Description

Federally funded research data facilities such as those funded by the National Science Foundation and NASA or the European commission, serve specific communities within scientific domains. Their mission is driven by the needs of the community they serve.

In the guardianship funding model, the data facility may adopt and/or develop software to serve their community. The costs of maintaining the software would be incorporated in the operations and maintenance costs of the

data facility. Success of this model requires a strong advocacy from the community as well as within the data facility.

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the funding agency of the data facility and budgeted as part of operations and maintenance and all is managed by agencies.
Users – Not actively undertaken since support is provided by the requirements of the user community.
Communities– US Federal research data facilities are very well connected to their primary user communities and are dependent upon these communities for guidance (Board of Directors, Steering Committees, strategic planning).
Human Resources – Software and systems development and maintenance is planned, budgeted and implemented on an annual basis to meet the prioritized needs and objectives of the data facility and community it serves. Externally developed software may also be incorporated into the operations of the data facility.
Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the data facility, such as the central ICT unit, or more frequently is contracted to private companies.
Product management – Initial product planning and design may be undertaken through the support of a grant but long-term management is sustained through the data facility. The data facility manages the licensing, version control and distribution of the software.

Advantages

- long-lived, secure, well-maintained and are appropriately resourced for as long as the provision of data remains a government priority.
- facilities are well-known, visible, accessible and respected by the research community

Challenges

- Finding a home for software within a data facility that serves a specific community will be challenging if the software should serve numerous communities at once.
- typically, long-lived but not necessarily permanent.
- constant budgetary constraints and the value of software must remain high to justify continued support.
- The software requires a strong advocate and expertise within the data facility to be maintained.
- Facilities may wish to develop software in house (provided more control and in-depth knowledge of the code)

7.4.3.7 Software as a Service (SaaS) Business Model

Model Description:

A common model for software and services sustainability in the private sector is called "Software as a Service" or SaaS. While there are multiple definitions of SaaS, a common one that is useful for the present discussions is: "software that is owned, delivered and managed by one or more providers and is available remotely". The provider offers access to the software that is consumed in a one-to-many model at any time on a pay-for-use or subscription basis.

- Hosted
- On demand
- Integrated (operates on a platform)
- Subscription or other form of relation

- Multi–tenant (simultaneous use)
- Supports Network effect i.e. builds more rapidly, leveraging marginal benefit and tipping point

The initial service offering can be a free service that engages the community ("Freemium")
Typical Pricing models are:

Pricing Models (includes Freemium pattern)	.
Capacity-based Model	According to capacity, usage, or number of users reach certain thresholds
Feature-based Model	According to the number of key features available
Time-based Model	Subscription with a fixed (extendable) time
Use-case Model	Fees are based on specified categories of customers (non-commercial use, educational, non-profit, etc.)
Quality of Service Model	Customers get a certain level of response time or service level

Mapping Revenue Models onto Sustainment Attributes:

Funding – A complementary mix of contributions mentioned above.
Users –Large institutions and businesses that will benefit from the interoperability offered. Business model must address how to provide enough unique benefits from the service that customers will not simple download the software and apply their own software engineering.
Communities – For customers from the research community, community engagement through presentations at conferences, working sessions, demonstrations and other direct interactions are part of the culture. Community advocacy can support market development.
Human Resources – Can adopt support and contribution from user and supplier communities. Community contributions are done on a voluntary basis. For the research community, the mixed model of community contributions and a SaaS modality can be complementary and provide a broad customer engagement for sustainability.
Software Engineering – The role of software engineering is to develop and maintain an operational capability of the software.
Product management – "Traditional" SaaS implementations generally use conventional product management approaches. Documentation for users is usually web based and accessible through the internet. Chat rooms are a common means of offering support.

Advantages:

- cover a wide range of both open source and proprietary software capabilities.
- Flexibility of pricing from very modest costs to more structured implementations.
- Shorter development cycles when software can be pushed out to the cloud with smaller batch sizes, faster feedback and high overall quality.

Challenges:

- New capabilities may not be relevant to a subset of users and they must adapt to the change.
- The research community may be a low volume customer.
- Need for a set of software interfaces or application interfaces to provide a broad interoperability capability.

7.4.3.8 Information and Ad Sales

Model Description

Online information and ad sales are forms of marketing that use the Internet to deliver marketing messages to consumers. This could potentially be providing an additional revenue stream. This is how Google, Facebook, YouTube operate.

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the business or industry that benefits from information and Ads.
Users – Considerable effort may be required to identify businesses or industries that desire the information that can be obtained and to identify appropriate advertisers.
Communities – Caution must be exercised in selling user information.
Human Resources – Additional human resources may be required to capture and package data and information desired by external business and industry interests.
Software Engineering – Software development or contracts with third parties may be necessary to capture/package information and support advertisements.
Product Management –Requires significant relationship management.

Advantages

- automated mechanism and requires little human intervention
- scales easily as user numbers increase
- may include a user model where users pay to not see ads
- there is potential for annotation service provided by this model to become part of institutional quality assessment process
- this model can contribute to a business diversification strategy for resilience and growth

Challenges

- requires sales to advertisers and/or people that wish to use the service -- requires a sales team and an identified large consumer base
- requires good planning of communications and branding
- ads may disenfranchise user base
- sales of information is a major privacy issue and may disenfranchise user base
- questions about who owns the usage and annotation data

7.4.3.9 Corporate Support and Product /Service Sales

Model Description

Corporate Support –

The sponsorship model is one where corporate entities provide general or specific funding for the system as a whole or some subset of the system for which their sponsorship (and acknowledgement) is seen as being aligned with their mission or business model.
The membership model may provide more direct participation in system governance, depending upon the specifics of the model.

Product/Service Sales –

A product or service sales model is another approach for obtaining funds needed to sustain a software. In this instance, the business model includes an outright sale of software for customer platforms with the customer

having ownership of the software and then service to support operations on the customer's platform is provided by the vendor.

Mapping Revenue Models onto Sustainment Attributes

Funding– By potential corporate sponsors/members, requires a high degree of interaction and maintenance through time.
In the product/service sales model there is a requirement for marketing and promotion of the service as visibility and awareness of the service is a prerequisite for its use.
Users – Market development is a core component of both the corporate support and product/service sales models.
Communities – Engagement with the community of potential users is critical to the success of both the corporate and product/service sales models.
Human Resources – The revenue generated through corporate support would typically consist of negotiated funding amounts that are delivered over a specified period of time. The product/service sales model provides a revenue stream that is directly proportional to the number of items "sold" or delivered.
Software Engineering – In the case of both corporate support and product/service sales the system must have capabilities that are well aligned with the needs and requirements of the sponsoring organizations or customers.
Product management – The product management model must align with an overall platform that is flexible and extensible. This is required to meet the (often rapidly) evolving needs of the users/partners.

Advantages

- Capability to provide steady, long-term funding
- Can contribute to a business diversification strategy for resilience
- Could build partnerships with membership organizations to provide premium level service access

Challenges

- May require significant promotion and effective communication with corporate sponsors
- Expectations of "free" on the internet works against buy-in for the premium tier of freemium
- Tension between profit motive for Corporate partners and productive revenue capacity
- Challenge of differentiation between free service level and premium service level
- Association with some corporate sponsors may disenfranchise some users
- Providers of free data will not want distributors to profit from it if there is no added value

7.4.3.10 Consortium Model

Model Description

Organizations (Universities, non-profits, for-profits, foundations, individuals, etc.) establish a consortium based on a common mission to create efficient and enhanced use of resources. In this case, the Consortium provides a sustaining environment through provision of infrastructure, financial support, maintenance and community engagement.

Mapping Revenue Models onto Sustainment Attributes

Funding – Startup funding for the Consortium is often provided by a single entity (Foundation or other funding source). Continued support comes from Consortium membership fees.
Users – Significant ongoing effort may be required to identify and maintain the user community.
Communities – Community engagement is a significant effort in the consortium model and may require full time staff to maintain and increase membership.

Human Resources – The Consortium is typically governed by a Board of Directors who provide oversight and long-range planning.
Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the consortium body or may be subcontracted. Software may also be developed externally and adopted by the Consortium.
Product management – The software is managed to meet the needs of the Consortium and the communities it serves. Input to this process comes through the Consortium members.

7.4.4 Marketplace business models and strategies

The following marketplace business model clusters are adapted from Refs [v]:

7.4.4.1 Product community

Marketplaces help to build a community around products and this type of approach is relevant to digital goods. The marketplace primarily creates value to sellers and buyers by creating an active community of like-minded people. Users of these platforms are, for instance, enthusiastic about discussing and sharing specific content. To foster the community development, firms provide social network functions and focus their key activity on community building. Among revenue streams, commission fees are the dominant revenue form. If fees are differentiated, differentiation is most likely based on quantity. The cluster typically focus on one industry and operates globally. Overall, the 'product community' applies mechanisms and instruments that aim at increasing user engagement with the platform rather than pure transactions.

7.4.4.2 Offline services on-demand

Such marketplaces match service firms with consumers. The exchanged services are delivered offline and therefore require some form of scheduling. The primary value for both the businesses selling the services and the consumers demanding them can be related to their efficiency gains. For some firms, the value from brand image and platform design play an important role as well. None of the marketplaces creates significant value through the platform community. Rather, these firms focus their activities on generating solutions to increase efficiency. The companies in the cluster generate revenues through commissions from the sellers, while buyers mostly use the marketplace for free. Overall, this cluster resembles the concept of on-demand services. Therefore, we label the cluster as 'offline services on-demand'.

7.4.4.3 Online services

Companies in this cluster share the characteristic that they offer services that are delivered via the internet. This includes services that involve individuals 'sharing' their previously untapped skills. The cluster also includes firms that aggregate professional freelancers such as scientific researchers. These marketplaces provide a high efficiency to the supply side in earning an additional income or even to substitute their formal employment. Marketplace participation provides sellers with a clear advantage in reaching their target audience. In many cases, the demand-side users receive value from the community around the core service. In 75% of cluster firms, the value proposition is targeted at one single market (vertical). The marketplaces of this cluster monetize by charging sellers (68%), and/or buyers (20%). The fee is mostly charged as commission (55%) or subscription (28%).
The following strategies are adapted from Ref [vii]:

7.4.4.4 *Defensive One–Stop Shop*

Provider offers complementary products and services in a Marketplace to build a wide portfolio beyond the initial offering. This strategy leverages the client franchise and aims at increasing value per client by answering needs even outside of the legacy positioning. The digital Marketplace plays a key role to ensure the successful implementation of this strategy. The platform not only facilitates the products and services extension with limited investment.

7.4.4.5 Distribution Channel Extension:

The aim of this strategy is to develop new and complementary distribution channels via the marketplace without jeopardizing relationships with existing distributors and legacy channels.
Three main benefits arise from this strategy. First, the investment required is minimal. Second, it is a first step for the operator to reinforce relationships with distributors. Third, it improves client and business intelligence by providing access to additional data. In this model, the primary challenge is to foster digital channel adoption by clients. To succeed, MP operators need to support their existing distributors while avoiding to be seen as a threat to them.

7.4.4.6 Business Model Transformation

Players developing this strategy usually see their business at risk either because of upstream or downstream disintermediation or due to strong changes in their ecosystem, such as clients and suppliers going from captive to competitive. By transforming the business model, a player seeks to create value added solutions for both clients and vendors and build a sustainable position. The MP plays a crucial role in this transformation process by accelerating the transition and enabling a more flexible aggregation of services. Also, it provides an infrastructure platform to add new functionalities that enhance client and vendor experience. In order to succeed, the business requires three main elements. First, a successful reposition of its brand. Second, a decisive change management to grow focus on both clients and vendors and integrate service level delivery at the core of the company culture. Third, a swift time to market (TTM) to build a head start anticipating on forthcoming disruptions.

7.5 References

[i] Goldbeck, Gerhard. The scientific software industry: a general overview. Goldbeck Consulting, 2017. http://doi.org/10.5281/zenodo.260408

[ii] https://www.rd-alliance.org/group/brokering-governance-wg/outcomes/sustainable-business-models-brokering-middleware-support

[iii] Raymond, Eric. "The Magic Cauldron," The Cathedral and the Bazaar (Sebastopol, CA: O'Reilly, 1999), http://catb.org/~esr/writings/magic-cauldron/

[iv] Pearlman, Jay and Turner, Andrew. SaaS Business Model, RDA. 2015, accessed via https://www.rd-alliance.org/business-models-draft-saas-discussion.html

[v] Täuscher, Karl. Business Models in the Digital Economy: An Empirical Classification of Digital Marketplaces. Fraunhofer, 2015.

[vi] Makkonen, Juho. How to Choose the Right Marketplace Business Model. 2015. Accessed via https://www.sharetribe.com/academy/how-to-choose-the-right-business-model-for-your-marketplace/

[vii] de Panafieu, Olivier. THE SPRING OF B2B MARKETPLACES. Roland Berger, 2017.

Authors	Gerhard Goldbeck and Alexandra Simperler (both GLC)

Contributing partners	GRANTA, GCL

EC-Grant Agreement	723867
Project acronym	EMMC-CSA
Project title	European Materials Modelling Council - Network to capitalize on strong European position in materials modelling and to allow industry to reap the benefits
Instrument	CSA
Programme	HORIZON 2020
Client	European Commission
Start date of project	01 September 2016
Duration	36 months

Consortium		
TU WIEN	Technische Universität Wien	Austria
FRAUNHOFER	Fraunhofer Gesellschaft	Germany
GCL	Goldbeck Consulting Limited	United Kingdom
POLITO	Politecnico di Torino	Italy
UU	Uppsala Universitet	Sweden
DOW	Dow Benelux B.V.	Netherlands
EPFL	Ecole Polytechnique Federale de Lausanne	Switzerland
DPI	Dutch Polymer Institute	Netherlands
SINTEF	Stiftelsen SINTEF	Norway
ACCESS e.V.	ACCESS e.V.	Germany
HZG	Helmholtz-Zentrum Geesthacht Zentrum für Material- und Küstenforschung GMBH	Germany
MDS	Materials Design S.A.R.L	France
QW	QuantumWise A/S	Denmark
GRANTA	Granta Design LTD	United Kingdom
UOY	University of York	United Kingdom
SINTEF	SINTEF AS	Norway
UNIBO	ALMA MATER STUDIORUM – UNIVERSITA DI BOLOGNA	Italy
SYNOPSYS	Synopsys Denmark ApS	Denmark

Coordinator – Administrative information	
Project coordinator name	Nadja ADAMOVIC
Project coordinator organization name	TU WIEN
Address	TU WIEN I E366 ISAS I Gusshausstr. 27-29 I 1040 Vienna I Austria
Phone Numbers	+43 (0)699-1-923-4300
Email	nadja.adamovic@tuwien.ac.at
Project web-sites & other access points	https://emmc.info/