Fisheries New Zealand
Tini a Tangaroa

Frameworks for the next generation of general stock assessment models: 2019 CAPAM workshop report

New Zealand Fisheries Assessment Report 2020/39

S.D. Hoyle,
M.N. Maunder,
Z.T. A'mar

ISSN 1179-5352 (online)
ISBN 978-1-99-004329-1 (online)

November 2020

New Zealand Government

Requests for further copies should be directed to:

Publications Logistics Officer
Ministry for Primary Industries
PO Box 2526
WELLINGTON 6140

Email: brand@mpi.govt.nz
Telephone: 0800 00 83 33
Facsimile: 04-894 0300

This publication is also available on the Ministry for Primary Industries websites at:
http://www.mpi.govt.nz/news-and-resources/publications
http://fs.fish.govt.nz go to Document library/Research reports

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Hoyle, S.D.; Maunder, M.N.; A'mar, Z.T. (2020). Frameworks for the next generation of general stock assessment models: Report of the 2019 CAPAM workshop.

***New Zealand Fisheries Assessment Report 2020/39.* 80 p.**

The Center for the Advancement of Population Assessment Methodology (CAPAM) in collaboration with the National Institute of Water and Atmospheric Research Ltd. (NIWA) hosted a technical workshop on the creation of frameworks for the next generation of general stock assessment models (NextGen SAM) in Wellington, New Zealand, November 4–8, 2019. The meeting was attended in person by 82 people with another twelve attending remotely via Zoom. The 47 presentations and six discussion sessions were recorded and have been made available via the CAPAM website.

Contemporary integrated fish stock assessments are complex and require sophisticated tailored software, which can take substantial resources to develop, test, and maintain. Fourth generation languages (e.g., TMB) aid development but generating new applications remains expensive and highly technical. Unfortunately, there are too few resources in terms of funding and skilled stock assessment scientists to develop case-specific modelling software for all the populations that require assessment.

For this reason, general stock assessment modelling packages have been developed as tools to implement stock assessment models that can be easily applied to a wide range of fish stocks. General assessment packages have benefits beyond just the ease of application. A widely-used package provides a common language for scientists to discuss and present methods and results, provides a common platform for analysis and testing that improves reliability, gives confidence to stakeholders by presenting outcomes in a familiar format and in consistent manner between model iterations and with alternative assumptions and data, simplifies and enhances the peer review of implemented models, and allows development of a consistent set of 'best practice' guidelines for modelling approaches and assumptions.

In fisheries stock assessment, several general modelling packages (e.g., a4a, CASAL, Gadget, MULTIFAN-CL, SAM, Stock Synthesis) have been developed by disparate groups from regions with different management requirements, with little coordination between developers. Nevertheless, the resulting packages are very similar in approach and structure. However, due to differing data, underlying assumptions, and management objectives they do have significantly different features. For example, MULTFAN-CL was developed for tuna species and therefore includes modelling of spatial structure and tagging data, whereas SAM focuses on including modelling of random processes.

Well-designed stock assessment software should be developed to take advantage of new and innovative methods, data, and assumptions, and include the best features of current packages. To become widely used, it must be applicable across a wide range of fish stocks with differing data, assumptions, model structure, and management requirements. This model will require collaboration among a diverse set of scientists to take advantage of knowledge spread among the different disciplines, including:

- efficient estimation algorithms from mathematicians and statisticians;
- efficient programming from computer scientists;
- modelling assumptions from population dynamicists and biologists;
- likelihood functions from statisticians; and
- information and management requirements from fishery managers.

The goal of the workshop was not to develop the next generation stock assessment model, but to bring together a large and diverse group of skilled scientists to review the requirements and frameworks for the development of NextGen SAMs, to set the foundation for those interested in development, and to initiate collaborations to ensure success. The topics covered by the workshop included software development and coding philosophies; software architecture; the underlying language base; hosting of

software project development; stock assessment model features; user interface and good practice defaults; coordination and collaboration between scientists, institutes and agencies, and funders; software development project planning; and funding mechanisms for development.

Developing a NextGen SAM package will take time, and there will need to be a smooth transition from current models so that stock assessments resulting from such a package can be understood and trusted by scientists and stakeholders. The NextGen SAM should ultimately be able to recreate the vast majority of current stock assessments, excluding only extremely complex assessment models, research and development analyses, or case-specific stock assessments that require unique features.

A survey of the workshop participant developers and users of current stock assessment software established that there was a large amount of overlap in the capabilities of existing models and identified that there were considerable benefits from coordinating and merging the capabilities of the currently available general models. The survey also identified features that were vital to the NextGen SAM, including flexibility in specification of spatial structure, inclusion of tagging data, and statistical modelling of random effects. No existing model includes them all. Some desirable but less frequently used features can be left for future development (e.g., close kin observations and multi-species models), because the flexible software architecture should allow them to be added later. The NextGen SAM should also scale from data poor to data rich applications using the same framework without defaulting to ad hoc data poor approaches.

Technical aspects were highlighted in the discussions. For example, fully exploiting available computer hardware and processors (e.g., multi-core CPUs and GPUs) for parallelisation and concurrency at general and specific levels is vital to meet computational requirements to address common stock assessment needs (e.g., conducting diagnostics, ensemble modelling, and management strategy evaluation (MSE). Approaches for defining partitions (ways to group fish that have specific characteristics), and modelling transitions among partitions, are among the most important decisions when defining the structure of the NextGen SAM. The spatial resolution of the available data is becoming finer, allowing for finer spatial scale models, which requires rethinking spatial modelling within stock assessment models.

The NextGen SAM must be easy to use to ensure uptake and sustained support. It should also be easy to use appropriately, even by less experienced users. Therefore, the user interface is extremely important. Its main components are model setup, data entry, and displaying results. Model configuration should use an interpretable human-readable language so that users can read and interpret model configuration files to ensure the model structure, data, and assumptions implemented in the model are those that are intended or required.

Modelling methods and needs will continue to develop, and the model architecture must be designed so that new features can easily be included. This requirement makes a flexible and consistent underlying architecture paramount. The associated code should implement the model in a way that is as general and modular as possible, similar to how Casal2 has been structured.

The NextGen SAM should be created within a comprehensive modelling framework that includes all aspects required to conduct stock assessments ranging from data input and user control, diagnostics and output reporting, and MSE. Some of these aspects will be part of the general model software, whereas others may be supplementary software that interfaces with the base code.

There was consensus at the workshop that the NextGen SAM should be a collaborative open-source project to maximise its capabilities and encourage wide-spread use. This approach requires a high level of coordination, management, and planning to make it successful. Modern professional standards of software architecture, design, development, and implementation will be important to its success. There is a strong argument that the project manager should be an experienced specialist software development manager, rather than a stock assessment scientist or science manager. However, the ideal project manager would also have scientific domain expertise. Successful software development frameworks (e.g., Agile, DevOps), and software that facilitates their application, are available, and these should be used. There should be an emphasis on using automation as much as possible with integrated testing at multiple levels designed early in the process before development starts. One of the first steps in developing the NextGen SAM is creating a specification document that can be used as the basis for developing the application.

The main impediment to developing the NextGen SAM is sourcing funding and coordinating a large project team. It would be a large project which requires dedicated funding and resources in the short term to develop the model, and sustainable funding in the long term to maintain and update the software. Given funding constraints within single agencies, pooling funding from multiple agencies should be considered.

1. INTRODUCTION

In Wellington, New Zealand from November 4–8, 2019, the Center for the Advancement of Population Assessment Methodology (CAPAM) in collaboration with the National Institute of Water and Atmospheric Research Ltd (NIWA) hosted a technical workshop on the creation of frameworks for the next generation of general stock assessment models.

Contemporary integrated fish stock assessments are complex and require sophisticated software, which takes substantial resources to develop. Fourth generation frameworks (e.g., Template Model Builder, TMB, see Kristensen et al. (2016) aid development of these models, but generating new applications remains expensive and highly technical. There are too few resources in terms of funding and skilled scientists to develop case-specific modelling software for all the populations that require assessment.

For this reason, several general models have been developed that can be applied in a wide range of situations. General models have benefits beyond ease of application. A common model provides a common language for scientists to discuss methods and results. Comprehensive development and testing improve reliability, give confidence to stakeholders, simplify review, and can result in standard guidelines for model assumptions.

In fisheries stock assessment, several general models (e.g., a4a (assessment for all, Jardim et al. 2014), CASAL (C++ algorithmic stock assessment laboratory, Bull et al. 2003), Casal2 (Doonan et al. 2016), Gadget (Globally applicable Area-Disaggregated General Ecosystem Toolbox, Begley and Howell 2004), MULTIFAN-CL (Fournier et al. 1998), SAM (state-space assessment model, Nielsen and Berg 2014, Berg and Nielsen 2016), Stock Synthesis (Methot and Wetzel 2013)) have been funded and developed by disparate groups with little coordination, but are nevertheless very similar in structure. However, due to differing objectives, they have some significantly different features, strengths, and weaknesses. For example, MULTFAN-CL was developed for tuna and therefore includes spatial structure and tagging data, whereas SAM focuses on random processes.

The term ‘next generation general stock assessment model’ (NextGen SAM) has been derived from the report ‘Implementing a Next Generation Stock Assessment Enterprise: An Update to NOAA Fisheries’ Stock Assessment Improvement Plan’ (Lynch et al. 2018). As they define it, next generation stock assessment has three main themes: expanding the scope to be more holistic and ecosystem-linked; allowing for new and innovative science for data inclusion and analysis; and a stock assessment process that is more timely, efficient, and effective at optimising available resources and delivering results.

When applied to a stock assessment package, the term has not been succinctly defined and is usually interpreted as a general stock assessment model that is more flexible, robust, and statistically sophisticated than current models, and can be easily applied to a wide range of stocks and scenarios. This report may help to provide a definition.

Development of the NextGen SAM will require collaboration among a diverse group of scientists to take advantage of knowledge spread among the different disciplines: efficient estimation algorithms from mathematicians and statisticians; efficient design and development from software engineers; modelling assumptions from population dynamicists and biologists; likelihood functions from statisticians; and information requirements from fishery managers.

The goal of the CAPAM workshop was not to develop the NextGen SAM, but to bring together a diverse range of researchers to discuss frameworks for next generation general stock assessment models, to set the foundation for those interested in development, and to initiate collaborations to ensure success. Topics covered included: coding philosophies and software structure, the underlying language base, hosting the project, stock assessment model features, user interface and good practice defaults, coordination, project planning, and funding. Scientists were encouraged to present work from both ongoing research and completed studies. Presentations were 20 minutes long with a 10-minute question period, with longer invited keynote presentations.

Most attendees at the CAPAM meeting on the NextGen SAM were stock assessment scientists rather than expert developers, but some developers were present. Presentations and discussions therefore focused more on features and requirements of future software rather than on the details of the implementation of current software.

A set of focus questions was provided to attendees in preparation for the meeting. These questions and a summary of responses are provided in Appendix 1.
Two surveys were distributed to attendees to help identify features that are a) included in current assessment models, and b) desired for the NextGen SAM. Responses are summarised in Appendix 2.

The report is based on discussions during the meeting, as interpreted and developed by the authors. The authors take responsibility for any misinterpretations. Credit for most of the ideas belongs to meeting attendees.

2. BACKGROUND

2.1 Requirements for a NextGen SAM

Pressure to provide management advice for more stocks worldwide is making substantial demands on stock assessment scientists. This can only be achieved by developing and applying new tools to allow a greater number of robust stock assessments to be conducted with existing resources. Currently, scientists use a diverse range of tools, some general and widely used, and others tailor-made for a specific stock. This is a relatively inefficient approach requiring considerable development time and software testing, and more coordination would better use existing resources to meet the needs of managers and policy makers. Meeting these needs will require many decisions, ranging from the general and strategic to the specific and tactical. Here we identify and, where possible, address, the issues that would be encountered.

What is required for future stock assessments? In general the requirement will be for a stock assessment model like Stock Synthesis (SS) that can be applied to most stocks that have adequate data; a modelling architecture like a4a that can be used to implement many different model structures; and a modelling framework like FLR (Kell et al. 2007) that accommodates a variety of stock assessment approaches, ranging from contemporary integrated statistical age-structured models to approaches used for data-limited stocks. Examples from current software packages give a useful insight into how a potential framework could be designed, and how it would need to be extended to include aspects such as management strategy evaluation. In fact, a general stock assessment model, with suitable architecture, within a consistent modelling framework will need to be considered to encompass existing and future requirements.

A typical ‘origin story’ for existing general models is for a single developer to implement a model for a specific stock and then apply it to more stocks, adding features as needed, perhaps with others contributing to development in an ad hoc manner. As models get more complex and more general, it is harder for the developer to maintain the code or ensure robust and reliable code, and a well-funded team approach is needed.

There are currently numerous stock assessment modelling applications available, ranging from widely-used general models (e.g. SS, Methot and Wetzel 2013) to specific code used for a single stock (e.g., the southern bluefin tuna assessment model; Butterworth et al. 2003). See Dichmont et al. (2016) for review of assessment packages used in the USA. Even within a single agency, multiple general models can be used in different regions. For example, the US National Marine Fisheries Service (NOAA-NMFS) has SS, BAM (Beaufort Assessment Model, Williams and Shertzer 2015), ASAP (age-structured assessment program, Legault and Restrepo 1998, Miller and Legault 2015), and AMAK (Assessment Model for Alaska, Anon 2015), which all do similar things. Moreover, there are alternative models currently in development. For example, MAS (Metapopulation Assessment System, Brodziak et al. 2017; https://github.com/nmfs-fish-tools/MAS) is being developed by NOAA-NMFS with a focus on spatial dynamics, environmental drivers, multi-model ensembles, simulation testing, and MSE. This degree of overlap is inefficient and some models currently in use only implement a subset of the capabilities available in more general packages.

General models used by everyone will reduce duplication and focus development effort. A range of options and features can be applied to each assessment without the need for recoding or extensive code testing. Using the same model makes it easier to communicate among scientists, allows stakeholders to get familiar with the stock assessments, reduces the risk of errors because the model has been run and checked by many people, and facilitates the review process (i.e., reviewers do not have to learn a new assessment model for every review and requests for models using alternative assumptions can easily be run without requiring new code development). It does not make sense to develop a custom model if there is a general model that can do the same thing.

Clearly, there should be coordination and merging of general models to achieve efficiency. However, developing a general model takes time, and there needs to be a smooth transition to a new general model so that the assessments can be understood and trusted by modellers and stakeholders. During the development of the NextGen SAM, existing models will need to be maintained. However, only essential development should be added to existing models and development effort should focus on the NextGen SAM. Other than for unique case-specific assessments or research development, the NextGen SAM should be able to recreate the currently available stock assessments.

However, there are some advantages of having different packages developed by different groups. Many scientific developments have arisen from scientists working on their own projects and taking different research paths. This may be hindered if activity was focused on a single general model, although this could be somewhat mitigated in an open-source project that facilitated side-projects in development branches. More importantly, multiple general models would safeguard against the risk of taking a suboptimal path during development, leading to underlying structure that cannot support features that may be required in the future. In some cases, features may simply not be possible to generalise and will always require a dedicated model. Finally, it is unlikely that everyone will agree on the same single general model either because of disagreement about the best scientific approach, ability to implement changes in a timely manner, the need for control over the codebase, decisions, or implementation, or the importance of academic credit.

Given the advantages of both a single general application and multiple developments, a hybrid approach seems appropriate. Different groups or teams would create their own general models, but encouragement should be given to having a single team to integrate these into one main general model. Efforts by other teams to develop and test new features should be encouraged and can be integrated into the main general model as appropriate.

This development of the next generation stock assessment model can be broken into 3 separate domains.

- The framework – The software framework and system that runs the models.
- The model definition – A unified and versatile method of defining the model and its parameters.
- The modelling methodology – How scientists use the package to build models and produce results.

The framework is an interpreter of the model and will determine the underlying architecture. It is therefore the most important component of the general model and needs careful consideration. It will determine what is possible to implement initially and as the model is further developed in the future. The model definition should be a simple and dynamic language that allows scientific staff to quickly and easily define the model they wish to run. Methodologies can be different for each scientific organisation. The NextGen SAM should not be biased towards any singular form of modelling paradigm.

2.2 NextGen SAM development

Modelling methods and the scientific requirements will continue to develop, and the model architecture needs to be able to easily include new features as they arise. This requirement makes a flexible underlying architecture paramount. The associated code should implement the model in a way that is as general and modular as possible. Essentially, we want to take ideas from the current suite of general models to the next level, while including modern software design frameworks and enabling ongoing future development.

Developing the NextGen SAM is a large project and may take several years for the development phase to complete. It will likely need to be developed in stages, but the outcome at each stage must be useful. For example, the first stage may just allow models with the capacity for general partitions (ways to group fish that have specific characteristics) but not implemented as areas, stock/species, or allow for tagging. Such a model could be applied to most current assessments. If well designed, then extending it to include multiple areas, stocks/species, or tagging data would be relatively efficient.

The first stage of development should emphasise the "model definition language" (MDL) with which scientists will define the specific model being implemented. For example, Casal2 uses a hierarchical framework to define a model and its children (and their children, etc.). This allows even the core concept of the model (e.g., age or length based, sex structure, and time periods) to be determined by the stock assessment scientist as part of the implemented model definition.

The second stage should involve designing and constructing a software architecture that implements the model definition language using an easily maintainable approach. Casal2 has done this by structuring the code to be very similar to the model definition language (e.g., top level objects in the model definition language, such as 'Process', are root level folders in the source code). The consistency in terminology between the user configuration definitions and the underlying code enables understanding of the codebase and facilitates the design and implementation of new features.

The development team should include software engineers, statisticians, biologists, stock assessment scientists, mathematicians, and policy makers. Scientists from these fields are involved with each stock assessment and its products. On the other hand, the larger and more diverse the team, the more difficult it is to coordinate input and make decisions to enable progress. The theme of team membership is further explored below.

Different aspects of the development will require different types of expertise. Defining the core requirements for input (model definition language) and outputs (report structure/formats) would be the biggest first step to success. One recommended approach is to keep the team to a minimum group of stakeholders with decision making power, each of whom represents a wider group of subject matter experts.

The main NextGen SAM should be implemented using a version control system (e.g., git) with one (sometimes more) production branch available to those intending to conduct assessments, and research and development branches that can be used to develop and implement new features. Features can be included in the production branch once they have been validated and tested. The branching strategy for the codebase is unlikely to be fixed and may change based on the number of people and the types of developer pipelines being used.

The NextGen SAM should be a collaborative open source project (https://opensource.org/osd-annotated), allowing for transparency in the underlying codebase and providing a mechanism for a wide range of scientists and developers to contribute to the project. This approach requires a high level of coordination, oversight, and planning to make it successful. Successful software development frameworks are available (e.g., Agile, DevOps), and software that facilitates their application, and these should be used to develop the NextGen SAM.

There should be an emphasis on using automation as much as possible. Human interaction within the process of merging, building, publishing and releasing package updates should be minimised where appropriate. Automated checks (e.g., code meets style and complexity guidelines, security bugs are identified, robustness and unit testing requirements are met, package builds across different operating systems, documentation has been updated, and examples of use included, etc.) should always be carried out. However, expert human oversight is vital when considering merging proposed changes into the production branch and will require significant resources.

A working group based at the US National Center for Ecological Analysis and Synthesis may provide an appropriate forum for organising and coordinating the NextGen SAM project. This was used for the ADMB project (http://www.admb-project.org/) that supports AD Model Builder.

Many factors are important for the project, including the leadership and governance structure, the project management approach, how to structure the long-term maintenance and support, how to facilitate and host the user and developer communications, what management software should be used, where should the codebase be hosted (e.g., GitHub), will it be based on existing modelling frameworks (e.g., TMB or ADMB), determining priorities for improvements or new features, and the underlying package framework.

2.3 Characteristics of the NextGen SAM

One of the first steps in developing the NextGen SAM is creating a specification document that can be used as the basis for planning the development of the application. The specification document can be separated into two components. The first component describes the overall framework and lists the high-level requirements, fleshes out essential details on these, and then prioritises them. For each, it determines the amount of complexity/effort required for completion. This part of the specification document will be flexible, be continuously updated, and responsive to the development process. The team developing this component of the specification document would ideally be kept to a few key decision makers who will act as representatives for stakeholders.

The second component of the specifications document will outline the features of the NextGen SAM, including the equations that define the implemented scientific methods. Before this component of the specifications document can be developed, the main characteristics that will be available need to be outlined as high-level requirements, such as whether it is age structured, details of the input and output data structures, what partitions will be needed, and the specific features (e.g., a double normal selectivity curve should be available).

The software architecture will determine the success of the model implementation and, to some extent, determine what it is possible to implement. Defining the MDL at a high level will have a trickle-down effect on all aspects of the framework. Defining the core rules of the MDL gives clear guidance as to how the architecture should be laid out. The MDL specification should allow flexible model construction so that analysts can easily develop case-specific stock assessment models as required.

A list of the desirable main characteristics can be created by comparing current general and specific stock assessment models and determining what characteristics are common among them or differ (for an example, see Table 1). In addition, a ‘needs’ statement should be created for each of the current models to determine additional features and, if it is a good candidate, as a baseline for the NextGen SAM.

In addition, the features of custom stock assessment models should be evaluated to identify what they do that current general models do not, whether they are required, and whether they can be implemented in a general model. The goal would be to synthesise all the good features being used, prioritising those that are most commonly used (or needed), but without limiting the addition of new features in the future.

Most contemporary stock assessment models include time and age structure, whereas some include size instead of age (e.g., for hard-to-age species such as crab or for those in which processes are size-based). Few include both size and age because of computational or data requirements. However, with increased computational resources, including both age and size in the NextGen SAM would be desirable. Ensuring that the model is efficient when either age and size are not being used is a necessity, which may or may not require parallel implementation of an age-structured model and a size-structured model to improve efficiency while allowing the use of common features (e.g., CASAL). The age, length, and time partitioning should allow flexibility for the user to customise the model size or time increments, which may be uneven.

Sex and life stage structure is also vital for the NextGen SAM because characteristics differ among the sexes at different life stages for many species. The functionality to change sex and life stage would also need to be included.

Growth platoon structure is available in SS, though not used often. This feature allows different groups of fish to have different growth rates over the life of the population in the model. It can be used to efficiently approximate a combined age-length structured model, in which length-based processes such as fishing mortality with size-dependent selectivity can change the distribution of length at age.

Modelling of spatial areas is required and has been implemented in many research models but has had limited use in practice because of computational demands and limited data to inform movement and area-specific abundance. However, as with age-length models, data availability and computational speed are constantly improving and inclusion of spatial areas in the NextGen SAM is vital. The approach to implement spatial structure and the required spatial resolution that is most appropriate has not been resolved for stock assessment models and would need additional research to inform best practice.

Most stock assessments include some temporal variation, usually in recruitment, but some also have included temporal variability in other parameters (e.g., selectivity, growth, or natural mortality). However, they typically implement the temporal variation in a penalised-likelihood framework, which may not be the most appropriate statistical approach. It would be preferable to implement temporal variation using random effects or a state-space framework (e.g., SAM or WHAM (the Woods Hole assessment model)), but this is computationally intensive. Given that improvements in algorithms and computers will likely make random effects practical, the NextGen SAM should include random effects.

Multi-species and ecosystem models have become common but are generally not used for management advice in the same way as stock assessment models. Nevertheless, there are several examples of multi-species models developed in a stock-assessment modelling framework. The Gadget general stock assessment model has the multi-species feature as one of its main foci. Including multiple species can improve the stock assessment by better representing recruitment and natural mortality as affected by biological interactions, and by sharing information through technical interactions (e.g., shared measures of fishing mortality). Modelling multiple stocks has similar considerations to modelling multiple species. One difference is that the catch of the same species from multiple stocks cannot be separated easily, although this can also be an issue with different species caught in the same fisheries. Technical interactions are somewhat better understood and less data-demanding than biological interactions. Therefore, a multi-species feature that addresses technical interactions might be desirable in the initial implementation of the NextGen SAM, while ensuring that the biological interactions can be added in the future.

The NextGen SAM should be able to fit to all the standard data used in stock assessments (e.g., catch, indices of abundance, and composition data). It should also be able to fit to data less commonly used in stock assessments, such as tagging data. As a special case, it should be able to fit to Close Kin tagging data, which may require additional features.

Many of the features discussed above (sex, stage, growth platoons, species, space) are based on partitions and this should be a main consideration in the development of the model architecture. There must be a general partition category that can be used to extend the model in the future if the need arises.

A wide variety of models can be approximated by an age-structured or size-structured model, including surplus production models and some data-poor methods. However, it is not the more complex model that is approximating the simpler model, but the other way around. Although there are some benefits of these simpler models (e.g., perform well in some circumstances under simulation, quick to run for simulation studies, applying simple models can often help analysts understand the data, etc.), the NextGen SAM should not include these simple models (surplus production models, delay difference models, most data poor methods) as separate applications, but should implement these models as closely as possible within its existing framework. This makes the assumptions more explicit and transparent, allows for alternative assumptions to easily be tested, and allows for inclusion of additional data that are not accommodated by the simple models. Some examples include using an age-structured production model to represent a surplus production model, fitting to age composition data and fixing parameters to represent a catch curve analysis, etc. If the architecture is modular, a simple model would not require the overhead of the capacity to implement a more complex model (i.e., run time should not suffer at the expense of generality). The supporting functionality of the NextGen SAM (e.g., input file structure, output file structure, diagnostics, graphics, etc.) should be available such that independent implementations for these simple models can also use this functionality.

Table 1: Some general stock assessment models and structural features currently implemented, focusing on random effects and partitions. Light green indicates that the model has the feature, and black that it does not. Mid-blue indicates that the feature is partly implemented.

Model	Random effects	Age	Length	Stock	Species	Sex	Area	Tag
a4a	Y	Y	l2a routine in FLR to convert catch-at-length to catch-at-age		N	N	N	N
AMAK	N	Y	N	Multiple stock version available	N	N	N	N
BAM	N	Y	N	N	N	N	N	N
CASAL	N	Y	Y	Y	N	Y	Y	Y
Casal2	N	Y	Platoons. Length-based in early development	Y	Y	Y	Y	Partly implemented
Gadget	N	Y	Y	Y	Y	Y	Y	Y
MAS	Needs improvement	Y	Partly implemented	N	N	Y	Y	N
Multifan-CL	N		N	Y	Y	Y	Y	Y
SAM	Y	Y	N	N	N	N	N	N
SS	N	Y	Platoons	As growth morph.	As growth morph.	Y	Y	Y
WHAM	Y	Y	N	N	N	N	N	N

2.4 Funding

The main impediment to development of the NextGen SAM is funding and coordination. It is a large project which needs substantial dedicated funding and resources in the short term to design and develop the model, and sustainable funding in the long term to maintain and support the software and add additional features. The model should be future proofed (the ability to extend it as required) and designed to take advantage of future developments in computing power. Budgets for most agencies are flat and demands are high, so getting funding is difficult. Pooling funding from multiple agencies should be considered. Sustainable funding is critical to retain staff and keep the project on track.

3. CODING PHILOSOPHIES, STRUCTURE, AND UNDERLYING LANGUAGE BASE

3.1 Introduction

The development of the NextGen SAM is a large task and needs to be planned and managed well, following a carefully considered software development strategy. The project should be structured as a "software development" project, although it will have characteristics of a scientific research project. The IT industry has decades of experience in successfully delivering software projects across many domains. The design documents are important and will require frequent updating. They should be light, but detailed where necessary (e.g., specific formulas for model features), include a skeleton of the project as well as the design and development details needed for implementation. Tools and techniques should be used to provide documentation (e.g., code layout, code commenting, Doxygen, DevOps pipelines). A well-known convention in software development is that sixty to eighty percent of lifetime costs go towards maintenance and support (Canfora and Cimitile 2001). Planning should include minimisation of these costs.

The design document should address the foundational structure and methodologies, as well as the rationale for why decisions are made in favour of a specific approach. For example, do random effects need to be included initially or can they be added later? There is a philosophy in software development not to add any unnecessary complexity. However, when developing a NextGen SAM, potential complexity will have to be considered even if not added, because the model must be able to accommodate future needs. It is impossible to anticipate all future needs, so the design architecture must be highly adaptable. Adding complexity when required should not require extensive refactoring or rewriting. For example, multi-species biological interactions could be left out of the initial development, but the underlying structure related to general partitions must be implemented so that biological interactions can be added in future if necessary. It is important to be able to add features, but more important to be able to add them efficiently.

The software development process used by the project management team will affect the likelihood of project success. One traditional approach, known as the waterfall model, passes development through consecutive stages of conception, initiation, analysis, software design, implementation, testing, deployment, and maintenance. It emphasises planning, time schedules, target dates, and implementation of the whole system at one time. The rigidity of this approach is problematic and can lead to project failure. One criticism is that designers will not anticipate all future difficulties, and it is better to revise the design than to persist in a design that does not account for any newly discovered constraints, requirements, or problems (McConnell 1993). The limitations of this approach may be unfamiliar to managers in science institutions.

Agile has become a leading influence on software development practices in the last two decades, and aspects of it are likely to be suitable for developing the NextGen SAM. The approach was codified in the 'Agile Manifesto' (Beck et al. 2001, Ambler 2010) and are highlighted by the following concepts.

- Individuals and interactions over processes and tools. Tools and processes are important, but it is more important to have competent people working together effectively.

- Working software over comprehensive documentation. Good documentation is useful in helping people to understand how the software is built and how to use it, but the main point of development is to create software, not documentation.
- Customer collaboration over contract negotiation. A contract is important but is no substitute for working closely with customers to discover what they need.
- Responding to change over following a plan. A project plan is important, but it must not be too rigid to accommodate changes in technology or the environment, stakeholders' priorities, and people's understanding of the problem and its solution.

Most recent software development ‘frameworks’, such as DevOps, are influenced by Agile methods.

Amongst other things, Agile involves releasing working versions of the software at regular intervals. This shows users and funders the ongoing progress, permits feedback, and gives them confidence that development is proceeding successfully. Advances in stock assessment methods and computing technology are likely to occur during the software development cycle and will need to be taken into consideration. Agile development is designed to cope with this and allow the development team to adapt.

Choosing the software development framework to be used is itself a very important decision. The choice should be led by the software architect responsible for its implementation, with input from suitably experienced software engineers involved in the project.

Planning requires communication between software engineers and expert teams. Implementation is a large project that will require ongoing contributions from personnel in the roles of project manager, stock assessment expert, statistician, and software engineer. Fishery managers and other stakeholders should be involved as the ultimate ‘customers’ of the project. Experts should be used where appropriate (e.g., for GUI creation and back end development) rather than relying on non-experts. Preferably, the core development team will be collocated to allow face-to-face meetings (or will meet regularly), but there will undoubtedly be other contributors and a distributed model will be required to manage the project. A comprehensive code review process and coding standards are needed for the code to ensure that only fully verified changes are made, as well as the use of a robust continuous integration/continuous delivery system (CI/CD) to automate building, testing, and producing deliverables

Rollout is likely to be gradual as the NextGen SAM is applied to different assessments, and bugs are identified by early users and fixed. For the user, switching to a new SAM involves costs and risks, and change is always a hurdle in an environment with tight budgets and heavily scrutinised results: ‘if it ain’t broke, don’t fix it’. Users can be attracted by better functionality, a short learning curve, and institutional support. The first version of the NextGen SAM need not replicate all existing SS and CASAL assessments. It will be more important initially to focus on replicating and improving on a few high-profile assessments, to build confidence among users. High profile assessments will have the budget for parallel assessments with old and new software. Development should focus on structure, reliability, and speed rather than including a comprehensive array of options.

It may be difficult for the NextGen SAM to keep up with the latest methods in stock assessment. Research often progresses faster than model specification, design, and development. However, the application does not need to include the latest methods in the main production branch. There can be many experimental branches that are slowly included in development branches, and once validated and tested can be included in the main production branch. Demonstrated robustness and stability are more important for providing reliable management advice than including new methods.

Consideration should be given to developing “easy interfaces” to the NextGen SAM. This may include bindings for Python or R allowing a stock assessment scientist to quickly and easily apply a new process written in Python or R into the model without changes to the base code.

If the NextGen SAM provides a suitable open-source and modular framework for researchers to build experimental branches, with tools for simulation and model testing, it may develop into a platform for stock assessment research. R (R Core Team 2019) has a highly modular open-source framework and has developed into one of the main platforms for statistical research.

3.2 NextGen SAM architecture and underlying structure

The architecture or underlying structure of the application is one of the most important aspects of the NextGen SAM to get right. It must be flexible enough to allow for possible future features, within reason, while retaining efficient run times.

The model code could be developed using a back end and a front end. The back end would be the underlying architecture and the front end would be how the model developer interacts with the code (this differs from the user interface of the general model).

Two of the most important issues to get right to ensure the NextGen SAM is adaptable and efficient are the partition structure (e.g., age, sex, stock, etc.) and parallel processing.

The partition concept defines how to break up the population dynamics structure into different subgroups. Some partitions will be included in most if not all models (e.g., age and/or size, sex) and others will be used much less frequently (e.g., area, species, tag cohort, growth platoons). The underlying implementation needs to have a generalised partition so that future needs can be met. However, it may also be useful to pre-define commonly used partitions, to make specifying the model more convenient and to create more efficient code. One approach could be to include a general partition concept in the back end and to implement specific partitions in the front end. Transitions between categories within a partition (e.g., area) may depend on which category of another partition type (e.g., age) a fish is in. This may require a hierarchical definition for partitions.

The increasing number of processors on a CPU, the availability of GPUs with many processors, and access to high-performance computing resources can allow upscaling to much larger models. However, to fully realise the benefits of simultaneous and parallel processing, the NextGen SAM needs to be designed with parallel processing at its core. Stock assessment scientists need to work with statisticians, mathematicians, and software engineers to fully explore opportunities to use parallel processing. Parallel coding is difficult and often architecture- and operating system-specific and must be woven into the numerical methods written by the scientists. To mitigate this problem, systems are being developed to take serial scientific code and automatically generate platform-specific parallel code as output.

Parallel processing can be used in many ways, from specific components of the population dynamics model to the underlying calculations for automatic differentiation. Some approaches may use less efficient algorithms or the communication among processes may slow down the calculations. For example, parallel processing using numerical methods could be used to calculate derivatives rather than automatic differentiation, but the numerical methods are less precise and may be less efficient even when parallelised. Some parallelisation could be automatic for every model, whereas other parallelisation options may be specific to a model structure or data type.

Some possibilities for processes and/or groupings to parallelise or make concurrent are:

- Tag cohorts
- Multiple stocks/species
- Areas
- Any partition (or the general partition)
- Evaluation of observation likelihoods
- Calculating management quantities
- MCMC (speculative chain execution)

- The objective function optimiser
- Ensemble models, MSE, simulations, and projections

One structure that might need more design consideration is the spatial structure. Most spatial stock assessment models use a few spatial areas with block transfer rates. However, other analyses use finer spatial scales with other forms of movement such as advection-diffusion (e.g., SPM) or some form of spatial temporal autocorrelation (e.g., VAST, Thorson 2019). Including the finer spatial scales may require a different underlying structure. For example, the general partition can be used to implement few large strata but may not be computationally efficient as the number of areas greatly expands. Methods that allow movement only to adjacent cells can limit movement unrealistically when cells are too small or time periods too long.

3.3 Model codebase and coding requirements

Several major decisions need to be made about the model code. First, should the code be compiled into an executable, should the code be compiled at run time (which can be done in TMB applications from within the R framework), should just-in-time compilation programming languages be used (e.g., Java, Python) so that only the necessary functionality for the specific stock assessment model is executed, or should some hybrid approach be used? The approach should allow user to add features as code. Compiling at run time allows the code to be customised to the specific problem, eliminating redundant code that could slow down the model. With compile at run time there will be a compromise between compile time added and run time which may be reduced. There is also the possibility of pre-compiling some code that is always used or not changed and linking using shared libraries. This decision will also be influenced by the execution environment of the application, either locally (on a desktop computer) or remotely (on a server). For example, Casal2 uses an object chain approach. The code goes through a pre-determined path that includes multiple checks (e.g., did the user specify X? if so, do Y). The user input is read in at the start and a chain of objects is created where each object calls the next object in the chain once it has finished execution. This approach requires more time at the start to create the 'runtime' representation of the model, and less time during execution. For larger (and longer) model runs this provides a speed benefit as less work has to be repeated.

Another decision is about the type of code used. C++ has commonly been used in stock assessment (e.g., ADMB and TMB). However, other programming languages or modelling frameworks may be better for developing different components of the NextGen SAM. For example, TMB is embedded in R to allow for more flexibility in using the code and manipulating input and output. (Also Stan, https://mc-stan.org/; Julia, https://julialang.org/). The initial base application could be written in a single language, but the flexibility to interface with other languages to add features may permit a wider range of people to contribute to the project. Although languages like R are highly flexible, they come with significant speed decreases (e.g., Python can be much slower than C++ running the same code). C/C++ are ideal languages still because these allow easy access to low-level functionality for great speed, as well as interoperability with almost every language (e.g., Python and R both run C under the hood). Perhaps, once established, the new features could be implemented in the base language if that is easier to maintain, test, and/or is more efficient.

The selection of programming language should consider interoperability with different scientific libraries including various automatic differentiation tools (e.g., CppAD, ADMB). For example, C++ templates could be used so that a variable type can be changed without changing the implementation. This may allow the NextGen SAM to use alternative base variable types. For example, there are several different versions of automatic differentiation libraries and they may have different strengths and weaknesses. This may also allow the application to be run without the automatic differentiation calculations when they are not needed.

Existing implementations could be considered for inclusion into an initial codebase for the NextGen SAM, although code in many existing SAMs does not follow modern coding practices and would be difficult to integrate into a modern software development environment. Amongst existing frameworks and applications that do follow modern approaches, three with potential to use as a base or within the base code are TMB, Casal2, and the Analytics Template Library (ATL; the gradient-based optimisation library used in MAS; https://github.com/msupernaw/ATL).

TMB and ATL are not general models but implement automatic differentiation and other computational efficiencies needed for the random effects feature. TMB uses a template system and may be difficult to unit test. Both Casal2 and MAS include automatic differentiation and implement a general model, but it is not clear if all or any of the code from these projects should be reused. The underlying structure of Casal2 may be a useful starting point for further extension. It was designed as a versatile modelling framework with a re-usable codebase. Core concepts of its model (e.g., age structure or length structure) can be defined by the user as part of their model definition. Casal2 was also created to support multiple automatic differentiation libraries (e.g., ADOL-C, CppAD, BetaDiff) with minimal changes to the codebase. It takes a modular approach to allow for use of modern techniques (e.g., threading) in long-running parts of the model without impacting the main model functions.

The application needs to use efficient implementations to ensure short run times. These include highly technical aspects such as using physically contiguous storage, differences between accessing and modifying array elements, concurrency and parallel processing, thread awareness, etc.

The NextGen SAM should be designed for scalability so that complex assessments can be implemented. In general, this will mean taking advantage of hardware and multiple processors (e.g., GPUs). Parallelisation and concurrency at general and specific levels should be considered. Specific implementations include: 1) tag cohorts; 2) stocks; 3) function evaluations for MCMC, optimisation, derivative calculation, etc.; and 4) calculating likelihood functions for different data sets. Parallelisation might be implemented at a general level using the partition concept (particularly the general partition). Turning on and off the parallel processing and choosing the number of processors should be an option since communication among processes may be more time-consuming than the time saved by using multiple processors.

3.4 Code modularity

The implementation for the NextGen SAM needs to be modular to allow multiple developers to add features without affecting the underlying architecture. Object-oriented software development goes with modularity. Modularity allows developers to break a problem down into smaller simpler problems, and to reuse code. The inputs and outputs of each module are easy to describe and unit test independent of other modules.

The approach to modularisation is an important aspect of the software architecture. If possible, the whole project from the underlying implementation to the user interface should be modularised. For example, the GUI and other applications should be able to directly call functions in the application/libraries.

A somewhat related example is the way TMB is implemented as an objective function, so that different R algorithms can call the objective function in different contexts (e.g., to maximise likelihoods or run MCMC).

Another example is R itself, which provides a platform (the R base software) to which modules (packages) are added. Each module has defined inputs and outputs, with automated documentation (via roxygen), and testing (via internal processes and CRAN). The user decides which modules to use for the task at hand and can write additional code or their own modules.

3.5 Model testing and validation

Integrated testing at multiple levels is a standard part of modern software development. A large complex scientific software development project will require substantial effort to ensure that the implementation is correct and remains correct. An error- checking system should be designed early in the process before development starts.

Developers test their software in a variety of ways, from implementing a model component twice (by different people and/or in different languages) to simulation testing. Once a module has been accepted, testing it can be automated and carried out each time any of the application code is modified, even for minor modifications.

- Module unit tests. Known outputs from fixed inputs are used to ensure the modules produce the correct answer. These tests should include a wide range of inputs for the module being tested. Use of tools that determine what lines in the code have been tested might be useful to ensure complete testing of the code, as well as code coverage, profiling, and memory leak detection tools.
- Integration tests. Testing the interactions between units.
- Reference assessments / end-to-end testing / functional testing. Multiple stock assessments that use different features should be used as a set of reference assessments for which the results can be checked each time the code is modified. Testing all the features will be difficult. One approach is to set up a database of assessments such that all assessments accepted for management purposes are automatically included in the database and used for model testing.

The unit and system tests should be distributed with the NextGen SAM so that users can verify integrity of the system on their local machine. This is important because variances in CPUs/GPUs and Operating Systems can cause minor changes in results.

3.6 Developer requirements

Identifying the team to develop the application will be an important decision. Should developers be [scientific/research] software engineers or should stock assessment scientists be trained to use good software development practices? Who should be in the development team? What is the minimum diversity of the team? Should it include statisticians, biologists, etc? The team that develops the specification documents should represent all these groups, but those who write and manage the code will be more limited. In fact, the development group could simply be software engineers who interact with the project management team.

There was no consensus at the workshop on whether software engineers should do all the development, but there was consensus that modern professional standards of software architecture, design, development, implementation, testing, validation/verification, and documentation were vital. There is a strong argument that the project manager should be a specialist software development manager with considerable experience, rather than a stock assessment scientist or science manager. The ideal project manager would also have scientific domain expertise.

We also note that the feature specifications (e.g., the likelihood formulae, whether fish can change sex, etc.) are just part of the overall specification. The same features can be implemented in various ways. The ‘what’ is for the stock assessment scientist, but the ‘how’ is for the software architect and engineer.

Some argued that specific features (e.g., selectivity curves) could be developed by stock assessment scientists, perhaps as draft implementations to be refined later. It was also argued that each task should be done by experts in that field. Thus, technical aspects of the feature specification document would be written by stock assessment scientists, whereas the software would be designed and developed by software architects and software engineers. Gadget and Casal2 used software engineers to develop their code following feature specifications written by stock assessment scientists. The Casal2 software architecture was designed by a software developer with experience writing stock assessment software (SPM). However, stock assessment scientists will also need to be involved in design and development. And, to foresee issues, software engineers will need to understand how stock assessments work.

Once stock assessment scientists and other relevant experts have developed the feature specifications of the initial NextGen SAM, a team of scientific/research software engineers will develop the application.

The code should be designed so it is easy for anyone to modify the code to add additional features. Once this code has been tested, validated, and approved, then two options could be used. The first option is to allow anyone with the appropriate training to modify the code, which would go through a code review process before being accepted into the main branch of the code. The second option would be for a change to be proposed, perhaps in the form of code in the research branch of the code repository, which is then examined by project managers. If accepted, a specification document is developed, and software engineers efficiently implement the code and check the output against the original code developed in the research branch. These approaches would require separating the code into a main production branch and development branches – a standard approach for managing development. The latter approach would be flexible in the number of development branches. In contrast, the former option might result in conflicts if multiple software engineers are working on the same code in different development branches. The reason for allowing anyone to include code in the developmental branches is because inspiration often comes as stock assessment scientists work with their model implementations and learn as they refine the methods.

An alternative would be for software engineers to develop the back end and stock assessment scientists to develop the front end. This could reduce inefficiencies arising from stock assessment scientists using poor practices, although what is considered back end and front end may be blurry. Software engineers can make code efficient in ways that few stock assessment scientists are aware of. Difficulties may arise when the scientific number-crunching is the computational bottleneck, and the developer needs to understand the science to code it successfully.

If non-professional software developers and/or engineers contribute to the code, they should be suitably skilled. Project managers should judge whether potential contributors have sufficient knowledge of software development practices and the design of the NextGen SAM architecture and data structures. This would minimise issues related to bad software development practices and lack of documentation.

3.7 Free and open-open source (FOSS)

There was consensus at the workshop that the project should be free and open-source (FOSS; https://opensource.org/osd-annotated). Open-source projects have some impediments. For example, some governments have rules about licensing, open-source, private organisations or profit-making. Government departments may oppose allowing software that they fund to be open source and/or free. Therefore, institutional support for open source is vital.

Making projects open source requires effort to ensure that users can understand the code and to answer questions. One way to reduce this effort is to limit access to the code to those involved in developing the code rather than to the public at large. However, good documentation, etc., should already be part of the software development practices used for the project.

The choice of open-source licence needs consideration. Inclusion of third-party code affects what licence can be used. This might be a factor in what third-party libraries and tools are used.

Liability must also be considered, because the NextGen SAM will be used by many organisations for legally binding management advice and any errors could have financial consequences.

3.8 Ancillary code and software libraries

Consideration should be given to using pre-existing software libraries. For example, ADMB and TMB already have many features for developing complex nonlinear models and include automatic differentiation for optimisation and random effects. Other libraries might be useful, such as those that implement linear algebra. TMB uses the R environment so it can access all the R functions, but not within the function evaluation, which is implemented using template metaprogramming in C++. Should the underlying code be based on ADMB or TMB, should it be more generic using templates, or should a completely different underlying automatic differentiation codebase be used? One way to address this issue is to determine what changes would be needed to TMB to implement the required features and follow good software development practices.

The R software environment (R Core Team 2019) could be used for the front end and a programming language selected by software engineers used for the back end. The R tidyverse (Wickham 2017) packages might be a good approach to use to organise the code. Regardless, R should be the basis for the user interface, because there are many R packages useful for stock assessment model development and analysis, and most stock assessment scientists already use R. Other options might also need to be available for the user interface. Consideration should be given to using databases instead of flat text files for data input and output. This may make running multiple models (e.g., in MSE) more efficient because writing flat files is slow. However, using a database rather than a set of static files for input may make reproducibility and transparency more challenging.

3.8 Package hosting

A git-based host such as GitHub (https://github.com/) or GitLab (https://gitlab.com/) is probably the best platform to host the code because people are already familiar with the environment. There should be substantial automation for tasks such as automatic builds, unit tests, etc. Standard practices are encapsulated in concepts such as DevOps, “a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably” (https://www.atlassian.com/devops).

Consideration should be given to hosting the software executables on a server rather than (or in addition to) being PC-based. This removes the problem of having multiple executable files for different operating systems. However, confidentiality of the data may be a concern. Local operability will be also needed for locations where internet access is not always available, or data transfer costs are significant.

4. STOCK ASSESSMENT MODEL FEATURES

The selection of model features to include in the NextGen SAM is the most interesting part of the project for stock assessment scientists. A list of desirable features could easily be created by surveying current general models and assessments. However, this may produce an overwhelmingly large list of features, some of which are outdated, implemented for research only, or not practical. Here we discuss some features that should be considered for inclusion in the NextGen SAM.

4.1 NextGen SAM groups/partition structure

4.1.1 General considerations

Defining partitions, or ways to group fish that have specific characteristics, is one of the most important decisions that needs to be made when specifying the features of the NextGen SAM. These partitions range from those used in most models (e.g., sex, age/size, stage), to those used infrequently (e.g., species), and those that relate to the types of data used (e.g., tag cohort).

Transitions between partitions must also be defined. Some transitions are automatic (e.g., age), whereas others are not possible (e.g., species). Most transitions must be specified, parameterised, and the parameters estimated or fixed. Transitions within one partition type may depend on the state of another partition. Therefore, some form of partition hierarchy may be needed (e.g., see Casal2). Typically, there are many possible transitions and it becomes cumbersome to specify them, particularly if flexibility is desirable.

It may also be desirable to share parameters among partition categories, and data may need to be combined across partition categories. For example, when a sex-structured model is used to represent differences in growth, natural mortality, and selectivity between males and females, some data may not be available by sex. The transition matrix may also change over time. Therefore, it would be useful to develop a comprehensive way to model transitions, perhaps using a system like R uses for defining models.

Partitions can be modelled either explicitly or generally. For example, explicit partitions could be created for time, age, and sex to accommodate many stock assessments. Alternatively, a general partition could be created and used for time, age, and sex and the transition among partitions created to implement them appropriately. Explicit partitions may make it easier to understand parts of the code and easier to implement but may prevent expansion to new partition types in the future. MULTIFAN-CL uses the area partition as a general partition to model sex and species with the appropriate structure for transitions.

For example, Casal2 uses a model definition language to define the population structure and relationships among partitions. The structure is represented hierarchically as categories, so that a format defined as ‘sex.stage’ might have the categories (male + female) x (immature + mature). Casal2 has a general partition and uses hash (lookup) tables to assign transitions. Relationships and transitions involve multiple processes that belong to a few major types (e.g., recruitment, mortality, maturation). These processes define relationships between categories (e.g., recruitment into the immature category based on the biomass of female mature). For each process, observation type, and report, the model definition language defines which categories they work with and how they work with them.

```
@categories
format sex.stage
names male.immature male.mature female.immature female.mature # 4
total categories

@process maturation
type category_shift_by_rate
from stage=immature # expands to male.immature female.immature
to stage=mature # expands to male.mature female.mature
rate 0.01 0.02 # females mature at twice the rate of males
selectivity [type=constant; c=1]
```

The decision about what is explicit and what is general will depend on convenience of implementation and use and efficiency. The back end architecture may be designed using a general partition, whereas the front end implementations might use the general partition to implement explicit partitions.

Possible partitions include: time, age, size, sex, stage, area/region, growth platoon/morph, recruitment/settlement event, stock, species, tag cohort.

4.1.2 Spatial structure and inclusion of areas

Spatial stock assessments that model spatial structure and the location of a fish stock across different areas are becoming more common and must be accommodated in the NextGen SAM. Currently, most models use the fisheries-as-areas approach to approximate spatial structure. This approach allows selectivity to vary by area, but still assumes all fish to be part of a single fully mixed population. However, there is uncertainty about when this approximation is sufficient, and appropriate diagnostics still need to be developed. The alternative is to model separate subpopulations with movement between them.

The main considerations when modelling separate subpopulations in space are the spatial scale (resolution), the transition processes among areas, and how parameters change among areas.

Most stock assessment models use only a few areas and use block transfer to move fish around, but some such as SEAPODYM (Lehodey et al. 2008) and SPM (Dunn et al. 2015) use finer spatial scales and non-block transfer movement models. The MULTIFAN-CL movement process allows movement among all areas by defining movement only between adjacent areas, which reduces the number of parameters.

Movement rates may also change over time and be correlated with environmental conditions or change with other partitions such as age or sex. For example, SPM and SEAPODYM use environmental covariates to move fish around via preference functions that vary among partition levels. These models should be investigated to determine if the way they model space and movement would be practical for the NextGen SAM.

The spatial resolution of the available data is becoming finer, allowing for finer spatial scale models. Finer spatial scales increase computation time and potentially the number of transitions that need to be modelled, while reducing the amount of information available for each cell (or transition). Therefore, there is a trade-off among scale, computation time, and precision. Finer spatial scale might help to model fish with restricted adult movement and a larval dispersal stage, or to deal with tag mixing. In addition, spatial distribution and related characteristics (e.g., growth and movement rates) are likely to change along a cline rather than in discrete population subunits.

There are various ways to estimate movements between areas as the number of spatial cells increases and block transfer rates become increasingly difficult to estimate. Information sharing methods can be helpful, such as spatial deviates as random effects around a common shared rate, or a random walk to model a cline. Such approaches could be extended to have a spatial temporal correlation structure on either movement rates or distribution (somewhat similar to VAST), but either option is likely to be computationally intensive. Continuous time and space could also be considered but is likely to be intractable for a stock assessment model. Movement models to consider include advection and diffusion, Markovian, natal homing, migration, and entrainment.

Parameters may change spatially or be shared. For example, Western and Central Pacific Ocean and Indian Ocean tuna assessments share longline CPUE catchability among areas, which helps inform relative abundance levels and therefore constrains movement rates.

Of specific concern is how to model spatial growth variation. Does a fish that moves between areas keep its original growth trajectory (determined genetically or by the natal environment), take on the growth pattern of the new area (determined environmentally), or a combination of both? Stock partitions (growth platoons/morphs) and/or both size and age partitions might help to model spatial changes in growth. Constraints are needed to maintain realism: a fish should not change size simply due to moving to a new area with a different growth curve. Growth may be modelled using growth increments rather than taking growth directly from a growth curve.

Another concern is the stock-recruitment relationship and whether recruitment in an area should be a function of the spawning biomass in that area or of the whole stock. Area-specific stock-recruitment relationships complicate the equilibrium calculations used to determine initial conditions and management quantities.

Many other concepts need to be considered, including density-dependent growth and/or movement, and temporal changes in movement modelled using covariates.

4.1.3 The time period and time steps within a model

Contemporary stock assessments are dynamic by nature and therefore explicitly model time. Traditionally, they have used a fixed time step, usually year, but as more detailed data have become available, more flexibility has become desirable. Time defines when processes occur (e.g., recruitment) and when data are collected (e.g., length compositions, tag recoveries). The time definition is more important for some processes and data than others. For example, it is particularly important to get timing right when modelling length composition, because fish grow within a year. Aggregating length composition data over a year of continuous growth causes a mismatch between the data collection dates and the period used for the expectation (usually the middle of the year). This will create poor fits to the length composition data which is usually modelled as observation error whereas it is actually process error. A finer temporal scale will reduce error from this source. Components that influence the length composition include birthdate, seasonality of growth rates, and time of the composition sample. Time steps may need to be smaller for younger fish that grow faster than for the older fish. MULTIFAN-CL can use different time steps for different fisheries. For example, troll-caught South Pacific albacore length frequencies may be modelled monthly to capture modal progression, whereas other fisheries are modelled quarterly (Hoyle et al. 2012). These issues should all be considered in the NextGen SAM.

Flexibility and customisation need to be included in the time dimension. For example, the time step should be allowed to change over time to reduce computational demands in early years when data have less detail by using a coarser time step, and to accommodate more detailed data in later years using a finer time step. The model should have a seasonal structure, with a customisable number of seasons that can differ in length.

Including more time steps will increase model run time, and approximations may be required for efficiency. For example, SS assumes continuous mortality throughout a season and calculates abundance in the middle of the season but allows lengths to be calculated at any point in the season to create length compositions. This approach is useful for fast growing fish.

Within-year depletion provides an example of how time step flexibility is needed. Within-year depletion data need to be modelled at a fine time resolution but may only be available for part of the year and for some years. For computational efficiency, a coarse time step could be used when within-year depletion data are not available and a fine time step when they are. Implementing within-year depletion models with the NextGen SAM would allow years to be linked (e.g., for a stock-recruitment relationship) and inclusion of additional data (e.g., catch composition). The model could also have higher resolution processes/time steps embedded in lower resolution processes/time steps, e.g., the spring months in the most recent 20 years may have a finer time scale to account for faster seasonal growth rates revealed by weekly fishery composition data.

4.1.4 Length and/or age structured models

Either age or length structure is standard in most contemporary stock assessment models. Inclusion of both at once (age-length models) is rare because this is computationally intensive and has substantial data requirements. Age-structured models with approximations to full length structure have been used in some applications using growth platoons in SS or growth slices to reduce computational and data demands (see workshop abstract by McGarvey et al. ‘Will age-and-length based modelling permit broader application of the next-generation fishery assessment model?’ given in A3.3 of Appendix 3).

Resolutions of age and length bins should be flexible and appropriately matched with the temporal definitions for both the population dynamics and the data. Bin sizes, particularly for length-based models and compositional data in either age or length models, should be customisable to allow for variation with fish size, time period, and data source.

The NextGen SAM should efficiently implement either an age-based model or a length-based model using underlying architecture that includes both age and length. For example, CASAL implements age-based and length-based assessments independently within the same framework.

4.1.5 Multi-species models

Ecosystem-based fisheries management (EBFM) is gaining traction and requires analysts to provide relevant advice. Appropriate tools include full ecosystem models, stock assessment-like multi-species models, and models of intermediate complexity (MICE). It is unlikely that full ecosystem models can be efficiently included in the NextGen SAM, but the other approaches may be feasible. As well as providing EBFM advice, including multiple species in a stock assessment model might improve estimates for the species of interest based on its relationships with other species. For example, a multi-species model in which predation affects juvenile mortality may help to estimate the stock recruitment relationship.

Should multiple species be included in the NextGen SAM? At minimum, the underlying architecture should allow for this in future, via a general partition and the ability to include indices of abundance for other species (e.g., predators or prey) that can drive population processes (e.g., mortality or recruitment). Developers should consider including multiple species (or similarly, different stocks of the same species) with technical interactions so that information can be shared among species (e.g., growth, natural mortality, and relative fishing mortality) and data aggregated across species (e.g., catch recorded by species group). Biological interactions, which are more complex and require significantly more data (e.g., a diet matrix), could be left for future development.

4.2 Modelling of observations and associated likelihoods

Contemporary stock assessment models are conducted in a statistical framework to estimate the model parameters by fitting to observational data. Such models link the underlying population dynamics to the data through the likelihood function. In broad terms, these models predict the expectation of the data and the likelihood function represents the sampling variation. But the model can also encompass process variation and other process error. The model can vary in complexity from a simple one-to-one relationship with estimates from the population dynamics model to complex models that include many aspects of the population dynamics model (e.g., as used for tagging data). The NextGen SAM should be able to fit to all the common types of data (e.g., catch, indices of relative abundance, and composition (age, length, age-conditioned on length, weight)) as well as other less commonly available data types (e.g., mark-recapture, tagging growth increment, stock composition, stage).

4.2.1 Tagging

Tagging observations are becoming more common as a data type in stock assessments as computational resources improve. Tagged fish that have the same characteristics (e.g., release date, region) are grouped into cohorts and modelled separately so that the parameters related to those fish can be estimated (or fixed). Therefore, a partition for tag cohort is needed in the NextGen SAM. The tagged fish follow dynamics similar to the total population modelled by the stock assessment model, with the main differences being that recruitment is the release of a tagged fish and additional complications such as mixing periods, tag loss, tagging growth retardation, tagging related mortality, and tag detection/reporting rates. Flexible parameterisations will be important for some of these factors, such as reporting rates estimated from tag seeding that vary among fisheries and through time.

The various tag types may require different observation models. For example, archival tags provide many estimates of each tag's locations during the release period. These location estimates are correlated, and the temporal resolution is usually much finer than the temporal resolution of the assessment model. Simple approaches to resolve this discrepancy involve picking a location during each time period (e.g., where it is at the start of the time period or the modal location), but future research may develop a more appropriate method that requires a more sophisticated observation model. Approaches such as individual random effects or data weighting may be needed to deal with the correlated data. Other specialised tagging models such as catch-conditioned models also provide information on movement.

The NextGen SAM model should also be able to use information from external analyses of tagging data, such as probability distributions on movement rates or fishing mortality rates. In many cases it will be better to analyse tagging data at a finer spatial scale than the assessment model and introduce the information as a prior or penalty distribution.

4.2.2 Close Kin

Close Kin Mark Recapture (CKMR) (Bravington et al. 2016) is a relatively new technique that shows promise for improving stock assessments. It has benefits over conventional tags such as not being subject to reporting rates, tag loss, or tag related mortality. To date only a few studies have been conducted and research into its reliability is limited. CKMR has been integrated into the southern bluefin tuna stock assessment. The method is based on “recaptures” of related individuals such as parent-offspring pairs, siblings, and half siblings. The challenge is to calculate probabilities across different relationship types. The analysis requires much of the population dynamics already included in a stock assessment model, similar to integrating traditional mark-recapture observations into a stock assessment. Current CKMR models require individual birth dates (age is important) and survival rates (both fishing and natural mortality), so integration with a stock assessment model is beneficial. The model must be age- and sex-structured, ageing error must be considered, and the use of plus groups (as in most stock assessment models) is problematic.

4.2.3 Likelihoods and priors

For each observation type there are multiple possible sampling distribution assumptions. Errors also include unmodelled process error, model mis-specification, and correlated residuals. Therefore, a myriad of likelihood functions can be considered for fitting models to observational data. Data weighting and correlated residuals must also be considered. Different functions are appropriate in different circumstances, so the NextGen SAM should include a wide range of functionality and allow users to easily evaluate and/or include new likelihood functions and prior distributions. Multivariate likelihoods and priors would be particularly useful to allow for correlations. For example, composition estimates from a VAST model could be included using a multivariate likelihood (e.g., multiplicative logistic, multivariate lognormal) and the VAST estimate of the variance-covariance matrix; a multivariate prior could be set up for the growth parameters using estimates from an external analysis of length-at-age data; or a multivariate prior on movement and fishing mortality rates could be introduced from an external analysis of tagging data.

More research is needed on likelihood functions, and the NextGen SAM needs to be designed to accommodate future improvements. For example, the likelihood for composition observational data needs to be able to deal with zeros, include correlation, and estimate the variance and/or covariance scalars. Alternative likelihoods should be available, such as the conditional log-Gaussian Cox process which deals with zeros but is not commonly used. WHAM has a logistic normal with four alternative approaches for treating zeros. Likelihoods may also need to be robustified (e.g., using a *t*-distribution or a multivariate mixture distribution), to automatically deal with outliers.

4.3 Reporting and evaluating management quantities

Providing management advice is the main purpose of conducting fishery stock assessments, and management objectives differ among agencies. Therefore, the NextGen SAM should calculate a suite of management quantities ranging from reference points and projections to simulation testing and full closed-loop MSE. It should be straightforward for users to specify additional derived quantities for output as part of the management advice, so they can adapt the NextGen SAM outputs to the requirements of their management jurisdiction.

Complex functionality such as simulation testing and MSE could be integrated as part of the NextGen SAM, but it is more likely that the NextGen SAM will be used as part (e.g., operating model or estimating model) of a more broad-based software package that implements simulation testing or MSE (q.v. FLR).

4.3.1 Estimation of reference points

All the standard traditional reference points (e.g., F40% B_0, MSY, and spawner-per-recruit reference points) should be calculated. Other more recent reference points such as dynamic reference points and fishery impact plots should also be available. The NextGen SAM should include uncertainty in the calculation of reference points since many harvest control rules use probability statements.

There are some complications when estimating reference points. For example, calculations related to equilibrium conditions become problematic when there is movement and a plus group, and the plus group may have to be extended to much older ages to obtain accurate estimates. Similarly, movement and area-specific stock-recruitment relationships may need consideration.

4.3.2 Undertaking simulations, ensemble modelling, MSE, and projections

Management Strategy Evaluation and projections generally require simulations, a different concept from the estimation that is the main purpose of the stock assessment model. Projections can be approximated with estimation, but such approaches have complications when including both parameter and future process uncertainty in the projections. Bayesian estimation approaches may overcome some of these difficulties. Bootstrap approaches can also be used, but their statistical appropriateness is questionable.

Running a full closed-loop MSE is computationally intensive and it may not be feasible for the estimation model to be the stock assessment model used historically for management. As a result, MSEs are often run with simplified estimation models. Several components of an MSE are computationally intensive:
1) Conditioning the operating model, particularly because the operating model generally includes more uncertainty and structural complexity than a standard stock assessment model. Because the operating model needs to be conditioned on data to be realistic, it is essentially a stock assessment model and could be implemented in the NextGen SAM.
2) The estimation model is run many times within the MSE for each time step that the assessment is conducted and for each different state of nature (run of the operating model). The estimation model is fitted to the data each time to estimate the model parameters. Therefore, if the NextGen SAM is used as the estimating model it needs to be very efficient.
3) The operating model is also run many times to generate simulated data and it is usually more complex than the estimation model.

Some speed issues can be addressed by parallelising independent sequential processes and using high-performance computing, others by using databases and memory instead of disk-based text files. Use of profilers can help identify where the most computational time is being spent.

Including simulation capabilities inside the general stock assessment model (e.g., SS) has made simulation studies of model performance easier as well as making the general model simpler to use as both the operating model and the estimating model (data files have the right format). However, independent (but related) simulation frameworks have also been developed (e.g., ss3sim https://github.com/ss3sim/ss3sim).

4.4 Model outputs and diagnostics

Diagnostics are important for model development, to ensure that the model assumptions are met, to identify problems, and to point towards model improvements. Some diagnostics are not well developed for stock assessment models; however, there are many commonly used ones and some recent developments. Some diagnostics are quite complicated and time consuming to conduct (e.g., R_0 component likelihood profile, the production model diagnostic, retrospective analyses, and hindcasting) and should be part of the NextGen SAM. They could be generated by the NextGen SAM code or with companion software (e.g. r4ss, Taylor et al. 2013). The calculation of the diagnostics should be automatic, with the option to turn them off (or on) to reduce run times.

There are some alternatives to consider:

- Profile on current biomass rather than B_0.
- Conduct one step ahead predictions of the data (e.g., CPUE).
- More work needs to be done looking at composition residuals and plots made by time, age/length, and cohort to help diagnose model performance.
- Influence plots may also be useful.

4.5 Parameter estimation

Stock assessment models estimate parameters by fitting to observations. Maximum likelihood estimation is often used in stock assessment models, but Bayesian inference is also commonly typically with MCMC. Integration across some parameters in a random effects or state space framework, typically using the Laplace approximation in ADMB or TMB, is also used. In most cases, model fit is determined by optimising the objective function that includes the likelihoods for each set of observations, and potentially Jacobians for transformed parameters. The objective function could also include penalties to implement distributional assumptions (e.g., recruitment deviates) and/or priors if using Bayesian estimation.

4.5.1 Random effects

Random effects (or state-space models) are typically used to deal with temporal variation in model parameters in a statistically rigorous framework. Random effects can be used for other purposes such as modelling correlations in data (data weighting), imputing missing data, and sharing information across factors other than time (e.g., catchability or selectivity across fisheries, population dynamic and fishery process across stocks or species, across age or length in selectivity curves) while estimating the parameters that govern the amount of sharing (variance parameters, or smoothness parameters in splines).

Integration across the random effects is computationally intensive and is typically conducted using the Laplace approximation in ADMB or TMB or using Bayesian methods (e.g., MCMC). The Laplace approximation requires third order derivatives calculated efficiently and accurately, for which ADMB and TMB use automatic differentiation (in conjunction with numerical differentiation for the uncertainty calculations). If random effects are not used in TMB, there is no additional overhead. TMB is more efficient at implementing random effects than ADMB. The penalised likelihood approach used in many assessments (e.g., SS) to reduce computational demands has a likelihood that degenerates towards zero (i.e., the likelihood increases as the standard deviation approaches zero, but the likelihood is undefined at a standard deviation of zero). The local minimum for the random effect variance parameter, if it exists, is negatively biased. However, some iterative approaches can be used to improve the estimates.

At the CAPAM meeting there was consensus that random effects should be included in the NextGen SAM. However, it is not clear which optimisation software should be used to implement it (TMB, ATL, or something else). Some general models currently use the random effects/state-space framework

(Table 2). WHAM is in the early stages of development and does not yet have many features. SAM has been around for longer and has been used for many assessments but is limited to ICES-type stocks with good catch-at-age data. Many more features need to be added before SAM is generally applicable. Other more general packages (e.g., SS and Gadget) do not have the Laplace approximation built in and would require substantial modification to allow for random effects, which is probably not practical. The Casal2 architecture may be more amenable to including the automatic differentiation required to implement random effects. Exploring this possibility may be a high priority. The ATL library used in MAS calculates higher order derivatives and is therefore another option for implementing the NextGen SAM.

Random effects greatly increase the computational burden and therefore complex models such as multi-species models with biological interactions may not be feasible. These types of models may have to run without random effects, at least in the short term.

Random effects allow for the calculation of temporal variation and correlations, which is useful when doing projections. Random effects are also good for implementing correlations with environmental variables because they can appropriately parse out the variance components and can deal with missing covariate data.

Table 2: Major modifications needed to turn existing general models into the NextGen SAM.

Model	Changes	Practicality and potential
SS	Recode in TMB to include random effects	Complete rewrite required
Casal2	Implement Laplace approximation using the AD for higher level derivatives	Unclear, worth investigating
SAM	Increased functionality (e.g., length comp data, space)	Unclear, potential depends on application architecture
WHAM	Increased functionality	Unclear, potential depends on application architecture
Gadget	Recode in TMB	Complete rewrite required

4.5.2 Estimation of uncertainty

Estimation of uncertainty is an essential part of evaluating a model’s performance and providing management advice. There are various ways to estimate uncertainty ranging from efficient approximations (e.g., normal approximation using the estimated variance covariance matrix) to complex time-consuming methods (likelihood profile, bootstrap, and MCMC). All these methods should be included in the NextGen SAM.

Model weighting is also an important component of estimating uncertainty. Preferably, automatic weighting approaches should be included in the NexGen SAM, but they must appropriately deal with correlated residuals to provide accurate estimates of uncertainty.

In addition to parameter uncertainty, model selection and model averaging approaches should be considered for inclusion in the NextGen SAM. Model selection or model weighting metrics could be based on AIC, BIC, DIC, cross-validation (e.g., using cohorts in length composition data), hindcasting, or other methods. Data weighting needs to be carefully considered, given the pseudo-replication common in fisheries datasets. Ensemble modelling is becoming popular, but methods to combine models may be better dealt with using methods external to the NextGen SAM code. One alternative to the ensemble approach that combines different models that would be more amenable to the NextGen SAM is to create an uber model with more complexity than the typical stock assessment and estimate all the parameters. This may be a more defensible and transparent approach but estimating all parameters may not be feasible due to data limitations or computational demands. Reverse jump MCMC might facilitate the ensemble modelling.

Some other issues to consider are how to weight models that use different data. One fundamental problem is that we are not interested in predicting the observed data, but derived quantities. We generally assume that if the model fits the data well, then it estimates the derived quantities well, but this may not be the case.

4.6 User interface and good practice guidance

4.6.1 General

The NextGen SAM must be easy to use to ensure uptake and sustained support. It should be easy to use appropriately, even by inexperienced users. Therefore, the user interface is extremely important. Its main components are model setup, data entry, and displaying results.

GUIs have been developed for model setup for some projects (e.g., SS), but their use has been limited and they have often been left behind by development of the SAM application. GUIs can also become cumbersome with all the possible inputs and results and need to be well designed to be effective.

The R environment is increasingly used in stock assessments to interface between the model and the input and output data. R is well designed for manipulating data and statistical analysis for pre- and post-processing, and for graphing results. Dedicated R packages have been developed for several SAMs to manage data input, reporting, and model configuration. For example, SS has the comprehensive r4ss package (Taylor et al. 2013); https://cran.r-project.org/package=r4ss, see workshop abstract by Taylor 'Processing and exploring assessment model output: lessons learned from a decade of work on the r4ss package' in A3.8 of Appendix 3), MULTIFAN-CL has the R4MFCL package, and FLR (Kell et al. 2007) provides a comprehensive framework in R for stock assessment and management advice. R scripts are used to develop models and make changes. Automating model development using R scripts provides a record of every step in the process and allows the analyst to easily rerun analyses if required, such as when errors are found in data inputs, or at the next assessment round.

An example of a comprehensive framework based on R is the ICES Transparent Assessment Framework (TAF, see workshop abstract by Magnusson & Millar 'ICES Transparent Assessment Framework (TAF)' in A3.8 of Appendix 3). Data files, model scripts, and results are made available online, so that after peer review anyone can browse, download, and rerun the final assessment run on either their own computer or the ICES TAF server. The workflow is run from R, with scripts following a standard format. Files are held on GitHub, which provides a record of changes through time.

Other tools have been developed to interface with stock assessment results for various purposes. The r4ss package (Taylor et al. 2013) can create HTML files to display results, and this function is very widely used. SAM has a server-based web interface for both running models and viewing results (https://stockassessment.org/). MULTIFAN-CL has a dedicated viewer (the MULTIFAN-CL viewer (Bouyé 2003) for exploring results, which has been continuously updated and used for nearly 20 years. MULTIFAN-CL also has the tuna management simulator (TUMAS, Hoyle et al. 2011), an interface for managers to view the results of stock assessments and explore alternative management strategies.

Model configuration should use an interpretable human-readable syntax (q.v. Casal2) so it is easy to review and read a set of model configuration files and know what is being done. Consideration may be given to creating a GUI from the configuration files, so that once a model is built it is easy for others to use it and make changes (i.e., the GUI is created based on what has been implemented without any extra options that will confuse the user). A key advantage of this is that it avoids having to maintain a complex GUI in lock-step with the model. Use of the GUI should be optional, so that the model can be embedded in automated scripts or workflows.

User interfaces assist not only the assessment author, but also reviewers and stakeholders. A standard user interface familiar to reviewers allows them to quickly understand the assumptions being made. They can also go directly to the output displays they think are important without relying on the assessment author to present them. Reviewers can also run the model with different assumptions. For example, r4ss (particularly the HTML output files) has greatly sped up the review process for assessments using SS and allowed the reviews to be much more thorough. Similarly, stakeholders who are familiar with the user interface standard outputs typically better understand the model results.

It must be quick for users to create simple models, but we must also be mindful of misuse. Given the large number of options available in some general models (e.g., SS and Gadget) it might be useful to provide a way to easily set up an initial model (e.g., templates or an expert system), which can then be modified as needed. The initial model could be based on the Stock Assessment Good Practices Guide that is a goal of CAPAM but is not yet complete. SAM has a default configuration based on good practices that forms a starting place for assessment development. There must also be extensive error checks and warnings to ensure that the user is not turning on the wrong options.

Stock assessment scientists have traditionally used text files for data input, model setup, and result output. Each application has its own standard file formats. For example, MULTIFAN-CL uses the *.frq file for some aspects of model structure and for catch, effort, and size data; the *.tag file for tag release and recovery data; the *.ini file for other model structure; and the doitall.* file for most of the details of model setup. SS puts most data in a single file (usually *.dat), and most model setup in the control file (usually *.ctl). Both SS and MULTIFAN-CL output results in several text files, each with their own formats.

The input and output data formats are an implicit part of the user interface. The NextGen SAM should have a general, stable, and well-documented data format that includes metadata, so it can be translated from and to the data formats of other SAM software. This applies to both input and output data because there will be a need to compare NextGen SAM results with existing models, other general models, and future models. There should also be a standard for simulation data so a variety of models can be easily tested. Cross-testing will help error checking. This will also greatly assist collaboration among model developers (e.g., ICES TAF project). Defined standards for input and output files would allow software to process and display results (e.g., diagnostic tools) to be shared by different model developers.

Translators should be developed to convert from existing model input file formats to the NextGen SAM file format to ease transition to the new application. Reverse translators should also be considered for some existing applications so they can be used to test the NextGen SAM. The translators do not need to be integrated into the NextGen SAM but can be external tools. A common language to set up models would also be useful. For example, a4a uses the R formula format and Gadget has something similar for parameterising the models.

Most assessment models use flat text files to input data and output results. The use of text files has the important advantages of reproducibility, transparency, compatibility with version control systems, and generality but uses space inefficiently and can be slow to read, write, and search. Standard data formats such NetCDF or JSON, or within the R environment as in TMB, may be preferable to text formats. Modern config file "languages" such as YAML should be considered for the input model configuration format. Gadget links to databases using scripts. Use of databases to store data in a minimally processed form allows for flexibility when specifying the model. Routines can then be used to extract the data in the format required, aggregating as needed. A database system may also facilitate cloud computing by making data transfer more efficient. However, outputs and inputs should also optionally include text formats to simplify debugging during model development, and for archiving.

R is familiar to most assessment analysts and should be considered for the user interface. R is designed to be easy to share functions, which will greatly increase the features of the user interface, particularly for analysis and displaying results. R can also integrate other languages, allowing people with other development expertise to contribute. Other programming languages might be more efficient for particular tasks. R has tools such as the Tidyverse packages (Wickham 2017) (https://tidyverse.org/) and the Shiny package (Chang et al. 2019) (https://shiny.rstudio.com/) that might be useful. An R user interface for the NextGen SAM should probably be planned and built from scratch to ensure a consistent and efficient approach, taking ideas and approaches from existing user interfaces (e.g., r4ss, R4MFCL, FLR), particularly for post-processing and displaying results.

Some things should be done within the NextGen SAM and others within the user interface. To make the model code efficient, readable, and small, more should be done outside the SAM. On the other hand, it would be efficient and reliable to use functions within the stock assessment code to post-process results. For example, equilibrium calculations to calculate MSY could use the population dynamics function in the stock assessment. This would avoid rewriting the population dynamics function, with potential for error. The NextGen SAM should therefore allow the user interface to access its functions.

Keeping the user interface up to date with the stock assessment model code may be a continuous task that will require ongoing funding. This is important because users rely on the user interface and its features, so when it is broken assessments can be delayed. The code should also be backwards compatible because some assessments might be done using an old version of the stock assessment model, but the user wants to use features in the latest user interface. There must also be translators to take input files of previous versions of the NextGen SAM and update them to the latest version.

Model configuration and data input is probably the most important and complicated process in creating the model. All effort should be made to make this process easy and reduce errors. Some important considerations in model configuration and data input are listed below.

Aggregating estimates across partitions to correctly predict the data is complicated and difficult to do by hand. Therefore, this process needs to be made easy and efficient. Specifying lookup tables to aggregate the data (e.g., Gadget) would be difficult to do by hand, so automation is essential. Scripting should be used to keep a record. Scripts that allow inheritance from one model to another with only the changes explicit could facilitate assessment development. Gadget allows inheritance from one stock to another to facilitate developing stock assessments. R scripts serve a similar purpose for SS and MULTIFAN-CL. A system that keeps track of all the files that have been changed (e.g., make files) might be useful. SAM has a unique model number that allows the user to track down what was changed and by who.

Input files should be dynamic, like the Casal2 input files which are designed to be readable and do not require a fixed order. They also need to be able to change a model with scripted commands, rather than changing the configuration and/or data files directly. Design should allow for transparency, traceability, and repeatability.

Automated checks are needed for model setup and data inputs to minimise errors. There should be a comprehensive set of warnings that can help the assessment author to create the configuration and data files.

Reports generated by the NextGen SAM must be customisable to different management agencies' needs. For example, the r4ss HTML output files have greatly simplified reporting at the IATTC. The main output is provided in the assessment report and supporting information of no interest to many stakeholders (e.g., diagnostics plots) is available on a website.

4.6.2 Automating multiple runs

A stock assessment model is typically run not once but multiple times in the development of the assessment and for diagnostics (e.g., R_0 likelihood component profile, jitter runs, retrospective analysis, ensemble models), uncertainty estimates (profile likelihoods), sensitivity analyses, and management purposes (e.g., forecasts, simulation testing, MSE). These may need to be carried out with different model assumptions, greatly increasing the number of model runs. To facilitate and manage multiple runs, automation is essential.

To maintain flexibility the automation capability should be separate from the model itself. For example, Stock Synthesis and MULTIFAN-CL assessments are often automated using R scripts and the packages r4ss and R4MFCL.

Gadget has several features to allow automation that should be considered for the NextGen SAM:

- Inheritance allows definitions from one stock to be used for another. New model runs are put in a new directory, but only need to include the files that changed.
- Scripting is used to make changes in the control files rather than the user making the changes directly in the control files. Therefore, one only needs to read the script rather than the whole file. This makes identifying changes and comparing changes among different model runs more transparent, traceable, and repeatable.

5. DISCUSSION

The workshop covered a wide variety of topics related to developing the next generation of general stock assessment models and there were presentations and discussions on each topic. Although there was not consensus on everything (and that was not the objective of the workshop), many concepts were widely supported and much of this is captured within this report.

The characteristics of the NextGen SAM can be elucidated from looking at existing general models, models designed for specific stocks, and features desired by practising stock assessment scientists. The survey of participants showed that there is a great degree of overlap among existing general models and this is an inefficient use of resources. Clearly, there should be coordination and merging of general models.

Developing a general model takes time, and there will need to be a smooth transition to a new general model so that the assessments can be understood and trusted by stakeholders. During the development of the NextGen SAM, existing models will be maintained. However, only essential development should be added to existing models and effort should focus on the NextGen SAM. The NextGen SAM should ultimately be able to recreate all available stock assessments within reason (inappropriate, research, or extremely complex models could be excluded).

The survey identified several features that are vital to the NextGen model (spatial structure, tagging data, random effects), but no existing model includes them all and they would require substantial resources to implement. Some features (e.g., simultaneous use of both age and size structure) may be less desirable, but fundamental to implement in the initial stages. Other desirable features that would require substantial resources to implement may need to be left for future development (e.g., close kin, multi-species). The NextGen model should scale from data poor to data rich applications using the same framework without defaulting to ad hoc data-poor approaches.

The next generation of general stock assessment models should be created within a comprehensive modelling framework that includes all aspects required to conduct assessments ranging from data input and user control to diagnostics and MSE. Some of these aspects will be part of the general model software, and others may be supplementary software.

There was consensus at the workshop that the NextGen SAM should be a collaborative open-source project. This approach requires a high level of coordination, oversight, and planning to make it successful. Modern professional standards of software architecture, design, development, implementation, testing, validation/verification, and documentation were vital. There is a strong argument that the project manager should be a specialist software development manager with experience in successfully developing scientific software, rather than a stock assessment scientist or science manager. Successful software development frameworks (e.g., Agile, DevOps) and software that facilitates their application are available, and these should be used to develop the NextGen SAM. There should be an emphasis on using automating as much as possible with integrated testing at multiple levels designed early in the process before development starts. One of the first steps in developing the NextGen SAM is creating a specification document that can be used as the basis for developing the application.

The NextGen SAM must be easy to use to ensure uptake and sustained support. It should be easy to use appropriately, even by inexperienced users. Therefore, the user interface is extremely important. Its main components are model setup, data entry, and displaying results. Model configuration should use an interpretable human-readable language (q.v. Casal2) so it is easy to read a set of model configuration files and know what is being done.

Modelling methods and needs will continue to develop, and the model architecture needs to easily include new features. This requirement makes a flexible underlying architecture paramount. The associated code should implement the model in a way that is as general and modular as possible, similar to how Casal2 has been structured.

There were several technical aspects that were highlighted in the discussions. For example, fully exploiting hardware and multiple processors (e.g., multi-core CPUs and GPUs) for parallelisation and concurrency at general and specific levels is vital to achieve computational requirements to address common stock assessment needs (e.g., conducting diagnostics, ensemble modelling, MSE). Defining partitions, or ways to group fish that have specific characteristics, and transitions among them, is one of the most important decisions that needs to be made when specifying the features of the NextGen SAM. The spatial resolution of the available data is becoming finer, allowing for finer spatial scale models, which requires rethinking spatial modelling within stock assessment models.

The main impediment to developing the NextGen SAM is funding and coordination. It is a large project which needs substantial dedicated funding and resources in the short term to design and develop the model, and sustainable funding in the long term to maintain and support the software and add additional features. Pooling funding from multiple agencies should be considered.

We provide a summary of the key steps for development of NextGen SAM:

- A specification document that can be used as the basis for developing the application should be developed.
- The planned characteristics of the NextGen SAM should be based on existing general models, models designed for specific stocks, and features proposed by practicing stock assessment scientists.
- The plan should encompass a comprehensive modelling framework that includes all aspects required to conduct assessments, ranging from data input and user control to diagnostics and MSE.
- Aspects not part of the NextGen SAM base code should be identified to be implemented in supplementary software.
- The plan and specification documents should be high level and only detailed where necessary. Rigidity can get in the way of successful implementation.
- Model development timelines should project a smooth transition to a new general model.
- The NextGen SAM should be a collaborative open-source project.
- The NextGen SAM should use modern professional standards of software architecture, design, development, and implementation.
- Development should be led by experts in software development management.
- Initial development should emphasise the human-readable “model definition language” (MDL) with which scientists will define the model to be implemented. The MDL will shape the design of the model architecture.

- The model architecture should be designed to easily include new features.
- This requirement makes a flexible underlying architecture paramount. The associated code should implement the model in a way that is as general and modular as possible.
- There should be an emphasis on automating as much as possible with integrated testing at multiple levels designed early in the process, before development starts.
- Implementation should be able to fully exploit hardware and multiple processors (e.g., multi-core CPUs and GPUs) to allow parallelisation and concurrent model runs.
- Development of the NextGen SAM will require funding and coordination, including pooling funding from multiple agencies.

6. ACKNOWLEDGMENTS

Thanks to Alistair Dunn, Hilary Oliver, and Jennifer Devine for reviewing the manuscript. Funding for the workshop and this report was provided by Fisheries New Zealand and NIWA. This report was completed under Objective 3 of Fisheries New Zealand project MPI19304.

The workshop organisation was led by Mark Maunder (IATTC) who also chaired the meeting. Local organisation of the meeting was by Simon Hoyle, Jennifer Devine, Teresa A'mar, and Jenny McLean (NIWA). Juan Valero (IATTC) coordinated the website.

7. REFERENCES

Ambler, S.W. (2010). Agile Software Development. Encyclopedia of Software Engineering 1:29–46.

Anon. (2015). Assessment Model for Alaska. Description of GUI and instructions available from https://github.com/NMFS-toolbox/AMAK/blob/master/docs/AMAK%20Documentation.pdf.

Beck, K.; Beedle, M.; Van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R.C.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D. (2001). Manifesto for agile software development, http://www.agilemanifesto.org.

Begley, J.; Howell, D. (2004). An overview of Gadget, the globally applicable area-disaggregated general ecosystem toolbox. ICES Conference and Meeting Documents 2004/FF:13.

Berg, C. W.; Nielsen, A. (2016). Accounting for correlated observations in an age-based state-space stock assessment model. *ICES Journal of Marine Science 73*:1788–1797.

Bouyé, F. (2003). MULTIFAN-CL Viewer. Pacific Community, Nouméa.

Brandon, J.R.; Breiwick, J.M.; Punt, A.E.; Wade, P.R. (2007). Constructing a coherent joint prior while respecting biological realism: application to marine mammal stock assessments. *ICES Journal of Marine Science 64*:1085–1100.

Bravington, M. V.; Skaug, H.J.; Anderson, E.C. (2016). Close-kin mark-recapture. *Statistical Science 31*: 259–274.

Brodziak, J.; A'mar, T.; Supernaw, M. (2017). Designing a General MSE Framework for a Movement-Based Metapopulation Assessment System, https://afs.confex.com/afs/2017/meetingapp.cgi/Paper/29195. *In* 147th Annual Meeting of the American Fisheries Society.

Bull, B.; Francis, R.; Dunn, A.; McKenzie, A.; Gilbert, D.J.; Smith, M.H. (2003). CASAL (C++ algorithmic stock assessment laboratory): CASAL user manual v2.

Butterworth, D. S.; Ianelli, J.N.; Hilborn, R. (2003). A statistical model for stock assessment of southern bluefin tuna with temporal changes in selectivity. *African Journal of Marine Science 25*:331–361.

Canfora, G.; Cimitile, A. (2001). Software maintenance. *In* Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals, pp. 91–120 World Scientific.

Chang, W.; Cheng, J.; Allaire, J.; Xie, Y.; McPherson, J. (2019). Shiny: web application framework for R. R package version 1.1. 0; 2018.

Dichmont, C. M.; Deng, R.A.; Punt, A.E.; Brodziak, J.; Chang, Y.-J.; Cope, J.M.; Ianelli, J.N.; Legault, C.M.; Methot, R.D.Jr; Porch, C.E. (2016). A review of stock assessment packages in the United States. *Fisheries Research 183*: 447–460.

Doonan, I.; Large, K.; Dunn, A.; Rasmussen, S.; Marsh, C.; Mormede, S. (2016). Casal2: New Zealand's integrated population modelling tool. *Fisheries Research 183*: 498–505.

Dunn, A.; Rasmussen, S.; Mormede, S. (2015). Spatial Population Model User Manual. *NIWA Technical Report 138*. 208 p.

Fournier, D. A.; Hampton, J.; Sibert, J.R. (1998). MULTIFAN-CL: a length-based, age-structured model for fisheries stock assessment, with application to South Pacific albacore, *Thunnus alalunga*. *Canadian Journal of Fisheries and Aquatic Sciences 55*: 2105–2116.

Hoyle, S.; Hampton, J.; Davies, N. (2012). Stock assessment of albacore tuna in the South Pacific ocean. WCPFC-SC8-2012/SA-WP-4.

Hoyle, S. D.; Bouyé, F.; Harley, S.J. (2011). TUMAS: a tool to allow analysis of management options using WCPFC stock assessments, WCPFC-SC7-2011/MI-IP-01, *in* WCPFC Scientific Committee, 7th Regular Session, Pohnpei, Federated States of Micronesia.

Jardim, E.; Millar, C.P.; Mosqueira, I.; Scott, F.G; Osio, C.; Ferretti, M.; Alzorriz, N.; Orio, A. (2014). What if stock assessment is as simple as a linear model? The a4a initiative. *ICES Journal of Marine Science 72*: 232–236.

Kell, L.T.; Mosqueira, I.; Grosjean, P.; Fromentin, J.-M.; Garcia, D.; Hillary, R.; Jardim, E. ; Mardle, S.; Pastoors, M.; Poos, J. (2007). FLR: an open-source framework for the evaluation and development of management strategies. *ICES Journal of Marine Science 64*: 640–646.

Kristensen, K.; Nielsen, A.; Berg, C.W.; Skaug, H.J.; Bell, B. (2016). TMB: Automatic differentiation and laplace approximation. J*ournal of Statistical Software 70*: 1–21.

Legault, C. M.; Restrepo, V.R. (1998). A flexible forward age-structured assessment program. ICCAT. Col. Vol. Sci. Pap 49: 246–253.

Lehodey, P., I. Senina, and R. Murtugudde. 2008. A spatial ecosystem and populations dynamics model (SEAPODYM) – Modeling of tuna and tuna-like populations. *Progress in Oceanography 78*: 304–318.

Lynch, P.; Methot, R.; Link, J. (2018). Implementing a next generation stock assessment enterprise. An update to the NOAA fisheries stock assessment improvement plan. NOAA Tech. Memo. NMFS-F/SPO-183.

McConnell, S. (1993). Code Complete: a practical handbook of software construction. Microsoft Press, Redmond, Washington.

Methot, R.D.; Wetzel, C.R. (2013). Stock Synthesis: A biological and statistical framework for fish stock assessment and fishery management. *Fisheries Research 142*: 86–99.

Miller, T.J.; Legault, C.M. (2015). Technical details for ASAP version 4. Northeast Fisheries Science Center Reference Document 15-17. NOAA Fisheries, Woods Hole, MA.

Nielsen, A.; Berg, C.W. (2014). Estimation of time-varying selectivity in stock assessments using state-space models. *Fisheries Research 158*: 96–101.

R Core Team. (2019). R: A language and environment for statistical computing, version 3.3.1. R Foundation for Statistical Computing, Vienna, Austria.

Taylor, I.; Stewart, I.; Hicks, A.; Garrison, T.; Punt, A.; Wallace, J.; Wetzel, C.; Thorson, J.; Takeuchi, Y.; Monnahan, C. (2013). r4ss: R code for Stock Synthesis. R package version 1.

Thorson, J.T. (2019). Guidance for decisions using the Vector Autoregressive Spatio-Temporal (VAST) package in stock, ecosystem, habitat and climate assessments. *Fisheries Research 210*: 143–161.

Thorson, J.T.; Munch, S.B.; Cope, J.M.; Gao, J. (2017). Predicting life history parameters for all fishes worldwide. *Ecological Applications 27*: 2262–2276.

Wickham, H. (2017). The tidyverse. R package ver. 1.1.1.

Williams, E. H.; Shertzer, K.W. (2015). Technical documentation of the Beaufort Assessment Model (BAM), NOAA Technical Memorandum NMFS-SEFSC-671. National Marine Fisheries Service, NOAA Beaufort Laboratory, NC.

APPENDIX 1: FOCUS QUESTIONS

A1.1. Software coding philosophies and software structure

Should the next generation model be coded by stock assessment scientists or professionally trained computer programmers?

Currently, most general stock assessment packages have been coded by stock assessment scientists; few have been coded by professional software developers. The stock assessment scientists' backgrounds in programming have varied from self-taught to computer science degrees. However, code written by stock assessment scientists is often poorly written and documented when compared with professional software development standards. For a collaborative project to be successful, code needs to be well organised and documented. This either means using professional programmers or extensively training stock assessment scientists in software development practice.

On the one hand, it is uncertain whether training of stock assessment scientists will be cost effective. On the other hand, professional programmers who do not have stock assessment expertise would require comprehensive specification documents and testing procedures to ensure that the model is implemented correctly. Also, collaboration will be less direct because other contributors will also be limited to writing specification documents rather than hands-on coding. Another issue is that creativity and new ideas are often a consequence of a stock assessment scientist implementing and testing their own model, which will be lost if professional programmers are used. Nevertheless, it can be argued that the next generation stock assessment model should first be an efficient, flexible, and reliable tool for assessing stocks and providing management advice. This is more important than its role as a platform for developing new approaches.

There is a variety of opinions on this topic. Some think that the clear answer is both. It will take a village to build the NextGen SAM well. On the other hand, others think that training a stock assessment scientist is unlikely to be cost effective. For example, in a specific case, an attempt to develop complex Casal2 functionality by people with limited software development training resulted in an eightfold increase in execution time.

Going further up than a professional programmer you need to also consider Software Architecture as a discipline. The way the overall structure of a system is designed before implementing code will significantly affect the quality of the resulting package.

Casal2 is a system for making models. It can be thought of as a system that compiles and runs a model based on an input configuration file. A computer scientist wanting to add new functionality (i.e., a new population process) to Casal2 would only need to add the new source code files to undertake the process and link these to the object manager code. Once done, the linking of the new functionality within the package, including configuration file syntax, documentation of the syntax, and error reporting and handling is managed by the existing codebase.

Casal2 has been designed to separate the framework code from the modelling code. Casal2 provides a harness for scientists to contribute code in specific (easily accessed) locations with most of framework automatically provided.

Systems like Casal2 can be built to support the running of external scripts for computer scientists to easily write and verify code in a non-performant manner before having it written and optimised by a programmer. For example, in development is the capability to allow a scientist to write a simple python script that would act as a new custom process in Casal2 with no Casal2 code changes required. Once the computer scientist is happy with their python script, they could hand it off to a programmer with test cases for it to be recoded using optimised code to become a permanent part of the underlying package.

Casal2 has been designed to allow scientists to do science first, but this requires programmers to do programming.

There's an important distinction between the science of finding the best model among a well-understood set of options for a production assessment application used in management and the science of exploring new modelling approaches that may have benefit for future applications. The first case clearly benefits from a generalised model that is efficient and well tested. The latter benefits from the scientist having the ability to directly change the model in fundamental ways that are deeper than adding a new functional form for some process. It's not necessary to use the same model for both types of science, but those exploring new approaches typically need to have a solid grounding in production assessments for their research to be grounded in reality, and there are inefficiencies associated with having a separate platform for productions assessments and exploratory research.

One approach is hiring professional programmers for short-term contracts without having professional expertise available on an ongoing basis. These contracts could be via the original developers or others that have adequate understanding to change and improve the software. However, this incurs the overhead of having to retrain new professional developers in the language and concepts of the underlying package.

How can we ensure that with all the desired features included the code remains computationally efficient for models that do not use the more computationally demanding features?

The more complex the features that are included, the more computationally intensive the calculations will become. For example, including random effects can make the models impractically computationally demanding and often results in convergence issues. Therefore, algorithms and code need to be as computationally efficient as possible. For some cases computational approximations may be needed, whereas the exact calculations may also be needed for simpler models or for evaluating the size of approximation errors. Alternatively, the addition of a feature may cause calculations to become more demanding even when the feature is not used, due to changes to the base code used for all models. In such cases redundant code may be needed such that one version of the code is applied if the feature is used, and another version if the feature is not used.

Professional programming is probably the pathway to achieving efficiency even when complex options are available. Computational efficiency is not the only relevant metric here. Experience with existing general models is that high diversity of options opens possibilities for users to select inappropriate combinations, or worse, for some combinations to create incorrect answers.

The architecture of the modelling system is very important here. A separation of the code that does the computations (science) from the code that builds the model is key. Casal2 compiles and runs a model based on the configuration file. No code that is not part of the model is used. All the processes in Casal2 have been written as a state-machine to ensure the least amount of code is executed at run time.

It might be worth considering code that compiles itself such that only the features that are desired are included in the model and then the code is compiled. This also may allow for additional features to be added efficiently at compile/run time.

Is there a coding standard that is appropriate for the development of the next generation model?

Most general stock assessment models have been developed without the use of a rigorous coding standard. Successful development of the next generation stock assessment model using a collaborative approach requires adoption of a coding standard to facilitate contributions by multiple coders as well as for continued maintenance and development. Coding standards are well developed in the field of computer science, and appropriate standards should be adopted.

Coding standards (how the code should be written) are a Programming 101 paradigm. Emphasis should be given to how unit tests are structured (e.g., to use a mocking framework, configuration files, integration tests, full test models etc). Casal2 uses the Google C++ coding style

Is there a way to easily allow the addition of new features?

The addition of new features to a general model generally requires adding code to the model and this greatly limits the number of people who can contribute. To facilitate the ability of assessment authors to add features, alternative approaches should be considered. For example, there are a variety of selectivity, growth, and stock recruitment curves that assessment authors might consider, and they may wish to test newly developed curves. If the general model had the capability to read in pseudocode that described the desired curve and to then interpret it into the underlying programming language of the general model (with or without recompilation), this would allow assessment authors to easily make simple changes to the general model. A simple example approach would be to allow the input of R-formatted GLM type code to link parameters to covariates.

Casal2 uses standard software architecture patterns. Adding a new object to Casal2 requires the addition of 2 lines of code into the relevant "factory" class. You can literally add a new nop (do nothing) process to Casal2 in 2 minutes, then focus on writing the actual scientific logic.

A1.2. Underlying language base

What language should be used for the basis of the general model?

General stock assessment models require efficient calculations to ensure that the complex models can be implemented. In general, the calculations involve optimising an objective function (e.g., the joint likelihood of the fit to the data) to estimate the model parameters. However, other options are also implemented such as Bayesian integration. ADMB has been the dominant language underlying many general models due to its use of automatic differentiation, which is efficient for optimisation. However, the recent trend towards state-space models (inclusion of random effects to represent process variation) has caused a move towards the use of TMB. Other versions of similar underlying programming languages have been developed and used for general stock assessment models (e.g., CASAL). ADMB and TMB are high level programming languages and based on other more standard programming languages (e.g., C++).

It is not clear if the next generation general model should be based on one of the existing programming languages used for stock assessment models (e.g., ADMB or TMB), an existing language not currently used for stock assessment, or a new language developed. There is potential for the next generation model to be highly computationally intensive in some applications, particularly if random effects are used. Therefore, further development of the underlying programming language may be required.

With consumer CPUs having 24+ cores now, there is a significant shift in paradigm for how software, especially high performance, is written. The use of multi-core CPUs and GPUs has provided a method of speeding calculations up by orders of magnitude. Automatic differentiation is typically a linear method and with newer CPUs the speed increase by using it is now lost when compared with multi-threaded algorithms.

New modelling systems should be multi-language. Your core engine should be written in a way that allows key people to optimise it against profiling (something like C++ is ideal here because it's so close to the CPU). Ease-of-use mechanics should be provided through interfaces with other languages (e.g., Lua, Python, Ruby, etc.) where execution can be handed off to slower code that is easier to write and iterate.

It's also worth considering the ability to find people with significant expertise in the languages being used. Python developers are very common.

A1.3. Hosting the project

What kind of code testing will ensure that the model has been implemented correctly?

Any next generation general model will require intensive testing and validation to ensure that the models are implemented correctly. Errors could be in both the specification documentation (if professional programmers are used) and in the actual code. Testing stock assessment models is complicated when results are not known in advance. This differs from testing where the errors are easy to see in the user interface, functionality, or well-established examples where calculations are simply repeated with different quantities. One approach that has been used is dual implementation in two completely different approaches (e.g., ADMB and Excel) where the same coding errors in both approaches is less likely. This works well for simple models, or subcomponents of moderately complex models, but may be difficult in a comprehensive next generation model.

One approach is to use case studies from existing stock assessment packages to test the next generation model. This approach requires exactly the same specifications among programs, with the same assumptions or approximations, so it requires the packages to be very similar. Also, new developments could not be tested this way and would require independent coding to test them. Another approach would be completely independent implementation by parallel development teams in the same programming language, which is common in commercial software development. However, this approach may be prohibitively expensive.

In addition to testing new features, old features need to be checked to make sure the new code does not break the old code. Therefore, there should be automatic testing (unit tests) using a set of comprehensive examples that cover the complete feature range to test against known results.

Casal2 has extensive unit tests covering the core platform. The core platform has significant code coverage (i.e., coverage of every line of code) with tests for many scenarios. Every build of Casal2 runs these unit tests to determine if the build is a success or failure. The scale of unit testing starts at testing a single line or equation with multiple known inputs and outputs all the way up to running multiple models through different run modes with expected outcomes.

Casal2 developers are also looking at putting in “stub-models”, specific models where the outcome isn’t 100% known but can be approximated (e.g., running MCMC to approximate Pi). Algorithms can be run against this to determine a measure of confidence.

New algorithms can be implemented and tests with known good inputs and outputs are used. Every single equation can be checked in isolation to ensure it’s consistent, increasing the chance that the sum of all pieces is equally consistent.

A1.4. Stock assessment model features

Do surplus production models and delay-difference model need to be explicitly included or is an age-structured model adequate?

Surplus production models and delay difference models are still used for stock assessment of several stocks. These approaches have the advantage that they do not need an understanding of the population dynamics subprocesses nor do they need catch composition data. In addition, it has been argued that these simple models can perform better in a management context than more complex models. Surplus production models can be approximated by an age structured model (although the age dynamics make the production curve change over time) and delay difference models can (probably) be represented exactly using an age-structured model. Therefore, is it necessary to also explicitly include production models and delay-difference models in a general stock assessment package? A separate package for these types of models could be used, but this would make comparisons between the simple models and the age-structured models less convenient because the outputs and diagnostics available in the general model would have to be developed for the independent package.

Does the general model need to include a VPA type approach?

Virtual Population Analysis (VPA) or cohort analysis is a common age-structured modelling approach to analysis of catch-at-age data. One main feature of VPA is that fishing mortality varies by age and year and catch (at age) is combined across all fisheries despite possible large differences in their selectivities. In contrast, statistical integrated stock assessment models separate catch into fisheries that have different selectivities and force structure on the selectivity (e.g., time invariant functional forms). VPAs can be approximated in integrated analysis by allowing temporal variation in the selectivity. At the extreme an independent parameter can be estimated for each age and year. However, this greatly increases the number of parameters and the possibility of convergence problems.

Having a VPA approach in the general model would greatly improve the ability to compare VPAs with integrated models. Another improvement would be to allow VPA calculations for some fisheries that have highly variable selectivity to reduce the number of parameters, while allowing restrictions on the selectivity for other fisheries or surveys.

One option is to allow very flexible time-varying age-based selectivity in an integrated assessment to approximate the VPA approach. Such a selectivity setup is a good idea in general, so could satisfy this need. However, a true VPA approach may be more efficient because it has no parameters and may have an analytical form to estimate the fishing mortalities.

Should data-poor methods be integrated into the next generation general stock assessment models?

There are a number of data-poor methods advocated for use when data for standard stock assessment models are unavailable or inadequate. These methods use a variety of data types and simplifying assumptions. Often the inherent assumptions are not immediately apparent. The underlying dynamics for a data-poor method should be the same as those used for a standard stock assessment model since these describe standard population dynamics. Therefore, arguably, it should be possible to represent a data-poor method with a standard stock assessment model. If not, then the assumptions inherent in the data-poor method may be unrealistic. If implemented in a stock assessment model the assumptions will be transparent. Also, alternative assumptions or available data could easily be investigated.

On the U.S. West Coast the comparison of purpose-built data-poor methods with standard stock assessment models was not hampered by assumptions about the dynamics, but rather the parameter sampling algorithm that was used in the data-poor model with only 3 or 4 parameters. In the end, the purpose-built data-poor method was reasonably replicated using MCMC with an integrated model.

What system should be used to decide what features are included in the next generation model?

A wide range of features could be included in a next generation stock assessment program, using a variety of approaches to implement them. Some ideas may be poorly developed, some may have a low priority, and some may simply be not very good. The features should be appropriate, and well tested and understood. Given limited resources, the features should be prioritised to determine which should be implemented first. However, the foundations need to be developed such that all possible future features can be implemented without a complete recoding of the foundations.

A system is needed to decide what features are included. This would probably involve a review panel to evaluate a recommended feature. Perhaps the concept would be recommended and if provisionally accepted, a full specification document would be produced and reviewed.

There may also be a need to have the capability to include experimental features, that may be limited in their distribution.

Any next generation assessment model should have some applications in mind when it is developed, some (if not all) of which should be stocks that are already being assessed using existing models. The features used in those existing applications would seem like a minimal set to include in the next generation model.

Does the general model need the capability to allow the distribution of length-at-age to change over time?

Most age-structured models assume that the distribution of length-at-age does not change over time. However, steep size-specific selectivity curves (e.g., escape gaps in pot fisheries and/or minimum legal size limits) and high exploitation rates can cause large changes in the distribution of length-at-age. The most flexible solution to allow for the length distribution to change over time is to explicitly model both age and length. However, this greatly increases the computational demands. Several approximations have been developed (e.g., multiple growth platoons, modelling the moments of the distributions), but their application has been limited. It is not clear what approaches should be implemented in the general model. Explicitly modelling both age and length might require specific changes to the foundations of the model.

It is possible that this can't be answered until approaches that allow length-at-age to change over time are applied to a large set of stocks to see how much difference it makes and whether the fit to the data is substantially improved. This would be straightforward to do with a large set of models that use Stock Synthesis; it just hasn't been done yet.

How can size-structured models be included in the general model?

Most assessments are based on age-structured models. However, there are a large number of assessments based on size-structured models. Size-structured models are typically applied to species such as crabs and lobster that are difficult to age. However, they would also be useful for species for which the population processes are predominantly size-based. The general model could be developed to be fully size- and age-structured and the relevant partition (age or size) eliminated/integrated out of the specific application. However, this may add too much additional computational complexity. Alternatively, separate age- and size-based models could be included in the code of the general model and the appropriate code used depending on whether the user chooses to use an age- or a size-based model. The other parts of the code will be used for each model type. Alternatively, a completely separate general size-based model could be developed, possibly sharing some of the subroutines from the age-based general model, where appropriate.

What time step should be used in the model?

The time step in the model should be flexible. Not just the step itself, but within a year, there should be customisable time steps that are of different lengths. One reason for this is that depletion estimators can be implemented for a part of the year when data are good, thus requiring short time steps, and larger time steps can be used for the rest of the year where the data do not support a depletion approach. Other reasons include seasonal changes in growth, natural mortality, and timing of recruitment.

Should parameters allow for density dependence?

Many stock assessments assume parameter values are constant over time except for recruitment. However, most, if not all, parameters probably change over time to some degree. There are many ways that parameter changes over time can me modelled. One particular case is density dependence, which may cause complications because it may cause interactions among the model parameters. There are also many decisions that need to be made such as which parameters can be density dependent, what component of the population is the measure of density (total biomass, spawning biomass, vulnerable biomass, cohort biomass). The density dependence may also be cannibalism and thus effect different stages in opposite ways.

Does the general model need multi-species capabilities, and should they be just technical interactions or include trophic interactions?

The ecosystem approach to fisheries management is becoming more popular and needs to be considered in stock assessment advice. Full ecosystem-based models are probably not practical in a general stock assessment program. However, multi-species models and models of intermediate complexity have been developed that are similar to standard stock assessment models. Technical interactions among species (different species caught in the same fishery) are much easier to implement than biological interactions (e.g., predation and competition). Biological interaction can be implemented in multiple ways ranging from simple indices of predation (perhaps implemented as a "fishery") to fully integrated multi-species models that fit to diet data. It is not clear how comprehensive the multi-species component of the model should be. However, if multi-species functionality is desirable, much time should be devoted to specifying the multi-species functionality before the foundations of the general model are developed to ensure that the desired functionality is feasible.

Should meta-analysis (Robin Hood) approaches be included in the general model?

Many stocks have limited data and assumptions need to be made to allow the development of an assessment and the consequent management advice. One approach is to share information from information-rich stocks with information-poor stocks. Typically, this has been done by meta-analysis of assessment results. However, doing the analysis outside the stock assessment model has some deficiencies in accurately representing the information and propagating the uncertainty. On the other hand, integrating multiple stocks and/or species into the same assessment model would greatly increase the complexity of the assessment. If the general model already has the capability of multi-species assessments, then adding the sharing of parameters makes sense.

Should the general model include automatic data weighting approaches for all data types?

Data weighting has become an important consideration in integrated fisheries stock assessment. Data weighting affects the relative influence of each dataset as well as the overall estimates of uncertainty. Similarly, the variance parameters of process error need to be determined. Ideally, the data weighting would be automatic. Unfortunately, the current approaches used for data weighting are less than adequate, often requiring ad hoc iterative procedures, not handling correlations well, need random effects/state-space modelling, and are not available for all data types. It is not clear what types of data weighting procedures should be available in the general model.

So long as data weighting remains an active area of research, it may be necessary for a general model to include both automatic data weighting approaches as well as less efficient iterative procedures to compare the results of the alternative methods.

Is the inclusion of multivariate likelihood functions necessary for the index, index composition, and catch composition data?

Most, if not all, general stock models assume that likelihoods for data components are independent. However, some data are correlated (annual values of a CPUE index) and new methods may imply correlations among data types (e.g., simultaneous spatio-temporal modelling of CPUE and composition data to generate index, index composition, and catch composition data). Such data sets would be more appropriately fitted using joint likelihood functions. This would require inputting the covariance matrices to use in the likelihood function. It is not clear whether using joint likelihood functions would improve the assessments.

Should a comprehensive set of likelihood functions be included?

There is a huge range of likelihood functions that could be included in a stock assessment model ranging from the standard lognormal and multinomial to robust likelihoods, self-weighting likelihoods, censored likelihoods, etc. Should also of these be included, just those that are considered appropriate, or should the model code be adaptable so that the user can create their own likelihood? One approach is to include as many as people desire or implement but to have an "expert system" user interface that selects the most appropriate, perhaps based on model results and diagnostics.

What type of facility for including prior information should be available in the general model?

Bayesian analysis has become common in fisheries stock assessment due to the ability to include prior information and represent uncertainty. The move towards integrated analysis has eliminated some need for data-based priors, but priors are still an important part of many assessments, even those that use maximum (penalised) likelihood and don't conduct full Bayesian integration. Therefore, the facility to include prior information is important to have in a general model. There is some overlap and similarities between priors and data, and questions about the statistical rigor of many approaches in fisheries stock assessment (e.g., priors on parameters or derived quantities, etc.).

However, some general approach to including prior information is needed. Decisions need to be made about which parameters or derived quantities priors can be applied to, and the forms of the priors. Specific attention might be needed to the correlation of prior information among parameters. For example, von Bertalanffy growth parameters estimated from an external analysis of age-length data will likely be correlated and this correlation should be propagated through the prior into the stock assessment for which the parameters will be updated from other information included in the stock assessment (e.g., the length composition data).

A general approach to allow priors with correlation among any model parameters might be desirable. This approach may be relatively straightforward to implement using a multivariate normal distribution, but the shape of the distribution may only be appropriate if the prior information is strong.

The FishLife project (Thorson et al. 2017) (see https://github.com/James-Thorson/FishLife) provides bivariate predictions of life-history traits (such as M and K) for all marine fish species in the world. An option to include bivariate priors would facilitate the use of these predictions as parameter priors.

In a marine mammal application, Brandon et al. (2007) showed that a bivariate prior was needed to avoid an incoherent joint prior distribution resulting from independent priors on two parameters.

However, such approaches seem unlikely to be adopted by more than a few people in the near term, so if this option is available, it would be important to not make it more complex to specify independent priors on individual parameters.

Should simulation and MSE features be included?

Simulation testing of stock assessment models and assumptions is a vital part of determining the appropriate model to use for a particular problem. This process requires simulation of both the population dynamics and the data gathered from the population. A related topic is MSE where the simulated population and data are used in a management loop to test a comprehensive management procedure that defines both the management action and the assessment methods. Some general packages have fully developed simulation capabilities, others have independent programs that are used, and others have both (e.g., SS and SS3sim). MSE is less well developed, but there are some general MSE packages that include a variety of assessment methods (e.g., Tom Carruthers' DLMtool and MSEtool, t.carruthers@oceans.ubc.ca). The obvious advantage of having simulation and MSE built into the general model is the convenience of having everything in one package and not needing to code the transfer of information from the estimation model to the simulation model and back. On the other hand, standalone simulators typically have more features and may allow for assumptions not included in the assessment model (e.g., data contamination, or different movement processes).

Operating models for MSE are often more complicated than the assessment model because they try to cover a wide range of sources of uncertainty, which suggests that an independent package for the simulator is preferable. Nevertheless, operating models should be conditioned on the data so that they are rational, which suggests integration into the general stock assessment model to force the good practice of using data-conditioned operating models. Both approaches could be taken, but this would result in duplication of effort. A decision should be made about which approach is to be used. Either approach should be adequate, and it is not clear which approach would be the most advantageous. Nevertheless, the general model needs to include simulation and MSE features.

The operating model could be based on a range of methods from the stock assessment model to finer spatial and temporal scales to agent based or individual based models. The more complicated the model the more difficult it is to determine if it is based on reasonable assumptions and if the combination of assumptions actually complement each other without data conditioning. Using the same model for the operating model as the estimation is useful to test specific model mis-specifications and other assumptions, whereas a more detailed model is useful for testing how the assessment model works in “reality”.

The inclusion of basic simulation and/or MSE features in a new platform does not limit anyone from developing independent simulators or MSE tools outside the model as well.

Is treating the future recruitments as parameters in a likelihood context adequate to model projections or should alternative approaches be used?

A computationally efficient method to do projections is to simply treat the projection period as part of the estimation period and estimate penalised recruitment deviates (or other time varying parameters). This could also be improved by integrating out the deviates as random effects. This allows for both parameter and initial state uncertainty and uncertainty about the future dynamics (the temporal variability in parameters). However, it is not clear if this correctly models the dynamics because it is based on the expected future and not individual possible trajectories (for example, the results will be around the expected stock recruitment curve not an individual trajectory of the stock recruitment curve). An alternative is to do stochastic projections, but this will ignore parameter uncertainty unless done in a Bayesian context, which requires priors on all the model parameters. Bootstrapping with stochastic projections has also been used, but this is intuitively ad hoc and “inconsistent”. Unfortunately, Bayesian methods are often computationally infeasible and more efficient algorithms are needed.

Can close kin methods be integrated into the general stock assessment model?

Close kin methods are becoming popular because they avoid some of the assumptions and issues with traditional tagging data and can provide an estimate of absolute abundance in a relatively short time scale. Close kin methods require many of the mechanisms in a population dynamics model so general stock assessment models may already have many of the mechanisms needed. (What additional features are needed?)

Are current integrated tagging models adequate or are finer spatial scales needed to deal with non-mixing?

Fully integrated tagging models have been around since the late 1990s, but they have not developed much over the past 20 years. The practicalities of tagging can often invalidate some of the assumptions made in the integrated tagging models. An important issue is that fish can rarely be tagged randomly, which can lead to incomplete mixing between the tagged and untagged populations. This is often dealt with by simply removing the recaptures in the first few time periods from the tagged population. However, this loses information, which may be particularly important for short lived species, and does not guarantee that the population is fully mixed after that time period. Finer spatial scale models can be used to better model the mixing, but these models may be too computationally intensive to include in a stock assessment model.

How should archival tag data be integrated into the general model?

Advances in technology are providing new sources of data. These data are often orders of magnitude more numerous in the number of observations than traditional data. This makes the integration of the data much more computationally intensive. In addition, the data are often correlated (e.g., in time and space) and this correlation needs to be appropriately dealt with. Archival tagging data is one technology that has potential to inform stock assessment models. The data could simply be aggregated into large time periods and/or space and used in a similar way that traditional tagging data are used, but this loses some of the information. Alternatively, the data could be used just for movement estimation. (What is the best way?)

What quantities should have estimates of uncertainty?

Uncertainty is an important concept taken into consideration by decision makers. Therefore, it should be presented for the quantities of interest. The quantities of interest can differ among applications and decision makers. Therefore, some flexible and customisable system is needed to determine which quantities the uncertainty is calculated for. In some approaches and underlying software, the calculations have to be specifically carried out (e.g., using automatic differentiation to calculate the variance covariance matrix) for a particular quantity, whereas others can produce estimates for all quantities simultaneously (e.g., MCMC). In the former, a more sophisticated approach may be needed to allow the model to be customised.

Is tagging data by predetermined age adequate or does the model need to convert from length to age?

An integrated tagging model is simpler to implement if the age of tagged individuals is known. However, this is not possible except using an approximation from the length of the fish. However, if growth is estimated in the stock assessment model then the age assignment should be updated simultaneously inside the stock assessment model. However, this greatly increases the computational demands of the model. (Is there a benefit?)

A1.5. User interface and good practice defaults

Should all possible features be included or just those that are considered good practice?

A huge variety of features could be included in a stock assessment model. For example, growth curves could be linear, von Bertalanffy, Richards, Schnute, growth cessation, Brody, Gompertz, etc., and each of these could have different parameterisations that have benefits in specific situations. In some cases, a flexible model could be used to allow implementation of the different options (e.g., the Schnute model can represent several other growth models with the parameters fixed at appropriate values). It is good to have a choice, but users often do not have the experience or knowledge to know which feature to use in a specific application. Therefore, it may be wise to only include those features that are known to perform well and include advice on when to use a specific feature. Also, having many features or options for a specific feature makes the model and its use cumbersome. Therefore, if all possible features are included, the user interface needs to be such that the model use is not cumbersome no matter how many features are available.

If only good practice or some expanded “not bad” practice is implemented, then a formal approach is needed to decide what represents good practice to include in the next generation model. There will always be a need to define a system to decide what should be included in the model, but a best practice approach will be somewhat different.

Casal2 by default has “nothing” for the model. Every aspect of the model is determined by the user using the configuration file to define their models. Casal2 has a concept of different states where a model can be sanity checked (not yet used). Consideration has been given to the use of model templates that create a known model structure and allow the user to specify only the parameters they wish to modify.

Some consider that the only way we can learn which practices are good or bad is through comparisons using models where multiple approaches are available. Even known bad practices can be useful to have available for transition from models which had no other option. It may be more important than ensuring only good practices are available to provide warnings or finding other useful ways to provide guidance to users when they attempt to setup models in ways that are widely understood to be not good.

What kinds of comprehensive user interfaces are worth the effort?

Most general stock assessment models have rudimentary user interfaces often based simply on text files. Others have more developed user interfaces. For example, SS has a GUI as well as a text file-based user interface. The GUI requires more resources to develop and maintain, and updating the GUI is often delayed. The majority of users use the text file-based interface. However, SS also has an R-based output interface (r4ss) with a browser display option, which is commonly used.

It is not clear that a GUI interface is beneficial but developing a GUI should be considered. Adding large amounts of data (e.g., composition data) is often cumbersome in a GUI and finding options to turn them on and off can be difficult, which is probably why many assessment authors prefer text files to GUIs. Assessment authors also tend to be more familiar with text files than GUIs. Perhaps most importantly, GUIs can lead to errors and are difficult to automate.

MULTIFAN-CL has an R-based package (R4MFCL) for manipulating both the text-based input files and the output files. It is commonly used for configuring assessment options in the input text files and setting up a series of assessment runs. Using an R script with tools from an R package as the user interface has the advantage of repeatability and automation, which reduces error and increases the potential number of runs. It also provides guidance when an earlier assessment is updated. This kind of user interface is arguably the most useful for stock assessments that are to be used for management.

One benefit of a well-developed user interface is the inclusion of error checking. Given the large number of options in a general model, it is easy to make errors (e.g., forgetting to turn on or off an option or getting data misaligned). Error checks may reduce the risk of these errors.

Good user interfaces also make the model more accessible to inexperienced stock assessment authors. However, this can also be problematic because inexperienced users may make poor choices when setting up a model and are more likely to make errors in the setup. Consideration should be given to implementing good practice defaults in the user interface.

Given the wide adoption of R, the next generation model should have an R based package to view results and diagnostics.

Python should be considered for developing the user interface. It is a language that is gaining great uptake in the scientific community because of the ease of use and ease of finding help. Python is also one of the core languages used in machine learning and AI/GPU based modelling systems.

Casal2 doesn’t have a GUI; text files are used, with an error system that identifies the file and line where the error occurred with a message of what the error was. Most errors are handled automatically by the system through an inbuilt parameter system (e.g., type conversion, invalid ranges, invalid values).

R is the current lingua franca of stock assessment scientists worldwide. Any model that does not have associated R functions to interact with it is not going to be widely used until someone inevitably develops an R package to fill that gap. However, there are many advantages to the use of plain text files as the fundamental input/output format to allow flexibility in the tools used to interact with the model, whether they are read or written by R, a GUI, Excel, or any other tool. Many assessment authors use text files rather than the user interfaces available for the general model to set up the model. However, the same authors typically use the (often R based) user interfaces to view results.

Should there be a comprehensive error checking system?

Due to the complex nature of general model and the vast range of features available, it is easy and common for assessment authors to make errors in setting up their models. It is highly desirable that comprehensive error checking systems are available to warn the user if the setup has potential errors that would affect the assessment results.

Casal2 has multiple levels of error checking available. A warning is displayed to the user when something may be an error. This does not stop the model execution but is displayed to the user. When errors occur from bad user input, Casal2 attempts to collect as many as possible before displaying them to the user. This prevents a one-error, one-run, one-error, one-run cycle. This makes it much easier and quicker for a user to diagnose and correct multiple or interrelated errors. Casal2 has a “Fatal Error” where the system must immediately quit and will notify the user of the problem. This is used when Casal2 is unable to collect multiple errors before finishing. Casal2 also has “Code Error” reporting. This is where something in the code is not consistent and is identified as a developer bug. A message containing information about the failure and a message to contact the developers is displayed.

A1.6. Coordination, project planning, and funding.

Should the code for a new general model should be open source?

The answer to this question is an obvious yes. However, there may be several decisions related to the open-source nature of the code. What open-source licence should be used? There may also be issues related to the funding source about whether the code can be open source, what type of open-source licence would it be, can anyone use it for commercial purposes, etc.

Is a next generation model necessary (as opposed to continued development of existing models)?

There are several general stock assessment models available that are consistently used for applications and are continuously being improved with additional features added. However, there are some features that are not available in most general models (e.g., integration over random effects using Laplace approximation) that may require substantial coding. Therefore, we are at the juncture where we need to decide whether to modify an existing package or program a new package from scratch. There are advantages and disadvantages to both. Further developing the existing packages avoids reprogramming all the currently available features, avoids users to having to learn how to use a new system, and retains confidence in the package that is already known. However, many current packages are poorly coded because they have been written by stock assessment scientists and not computer programmers, so recoding using coding standards offers many advantages. In addition, it is not certain if all the new desired features can be integrated into existing models. If resources can be combined and the necessary time spent planning the project, developing a new generation stock assessment model is probably the best way forward.

What factors have influenced the rate of adoption of existing models and what lessons can be learned from those experiences?

Among the different general model packages there has been a different rate of adoption. Some packages like Stock Synthesis have been widely adopted, whereas others like Coleraine have disappeared. There are many reasons some packages are more widely adopted, including: available features, developer’s responsiveness to modifications, availability of training, promotion, developer’s personal network, publications, ease of use. These should all be taken into consideration because widespread use of a package has many benefits, including: what factors have influenced the rate of adoption of existing models? and what lessons can be learned from those experiences? In the case of Stock Synthesis, widespread use has had clear benefits in improving the model in numerous ways including new ideas for features, bug reporting, justification of the investment in development, etc.

What characteristics will foster sustained contribution to the development and continued improvement of the next generation stock assessment model?

The success of any open-source collaborative software project depends on the sustained contribution of multiple developers. Some factors could include ease of contribution, recognition of contribution, ownership of the project, and available funding,

Is it necessary to be able to replicate results from existing stock assessment models to provide a bridge for users trying to make the transition to a new model?

Intuitively, these should be a requirement for any general model. In fact, on the U.S. West Coast, this is required in the terms of reference for stock assessments. However, not all existing stock assessment applications make sense under current knowledge or thinking. Therefore, the standard structure and features included in the next generation general model may not include all the possible options to represent existing applications. Adding these options would require additional work and possible overhead. This question is related to other focus questions that address whether only good practice should be included in the general model or options for “bad” or not so good practice also be included.

Should available resources be combined and dedicated to a single next generation model or should several next generation models be constructed by different groups?

Currently there are several general stock assessment packages available that were developed by different groups. These packages have features in common, but also differ in important ways. The differences are due to several reasons including the needs of the stocks being assessed, and the expertise and interests of the developers. For example, MULTIFAN-CL integrates tagging data because surveys are not available for tuna stocks and tagging data has become an important data type for some tuna stocks. Integrating tagging data into a stock assessment is complex and therefore only a few programs have this feature. However, any next generation model should have this capability to be globally applicable. Similarly, random effects, which is a dominant feature of SAM, is a vital feature of any next generation model. The generality of a package such as SS is also highly desirable. Therefore, should the resources from SS, MULTIFAN-CL, SAM, and Casal2 be combined to develop a model that is general, integrates tagging data (note that 3 of the 4 packages already do), and uses random effects, or should each of these programs be further developed to include the other features.

Despite the advantages of pooling resources in a single project, there are some advantages of multiple independent projects. Having multiple projects reduces the possibility of a fundamental flaw in the initial development of a project that may preclude an important feature in the future, because the flaw may not be in all projects. Also, developers may have different approaches and ideas leading to discovery of unique ideas that may not occur in a single project where creativity may be restricted by the thinking of the main developer. Also, cross comparison of different projects may help identify unknown specification or coding errors.

One approach may be to have a single next generation model, while encouraging independent projects based either on custom code or experimental branches of the main code. Once the ideas have been fully developed, they can then be considered for inclusion in the main code.

How should the next generation model be funded?

Acquiring adequate funding is vital for the success of any project to develop the next generation stock assessment model. Optimally, all funding would be available from a single reliable source. However, this may not be possible. Any other funding scheme has some risks that the funding does not become available or is delayed. Funding may need to be obtained from multiple sources and/or on an annual basis. Whatever the funding scheme, a rigorous budget would need to be developed to ensure that adequate funding is acquired.

Long-term ongoing funding will need to be secured, because general stock assessment models will always need maintenance and further development.

What project management system and software should be used to develop the next generation model?

Software projects require good management to ensure their success. For example, handling multiple developers, version control, experimental branches, committing changes, error checks, etc., all need to be dealt with. There are different approaches to software project management available and an appropriate one should be used.

There is no right answer to this because there are many different tools. The system chosen should not create friction for collaborators. It should be quick and easy for people who want to contribute to join. And everything that can be automated, should be. Casal2 runs 30 different build profiles when code is committed to the master branch. This is 15 Windows and 15 Linux builds.

Is there still a need for custom coded assessments if a comprehensive general model is available?

Today's stock assessments are a mixture of general model applications and custom code. Despite general model applications steadily becoming a higher proportion of the assessments, it is unlikely that custom coded assessments will be eliminated in the near future. A general model will never include all possible features, and the development and evaluation of new features may be easier in simplified code than in the code developed for the general model. Nevertheless, appropriate coding design may facilitate the development and testing of custom features within the general model code. So, although it is unlikely that custom coded assessment will completely disappear, there is potential for them to effectively disappear for official management use and to be relegated to the realms of research, if the general model is well supported and the additional of new features is rapid.

It should be noted that some of the differences between a general model and custom code are due to assumptions that are not necessarily the most appropriate, but more appropriate specification is available in the general model.

Custom code should not impact the model's capabilities. This is entirely dependent on how your software architecture is applied. Casal2 can have custom code added with no negative impacts on existing models or configurations.

How can we learn from other projects?

The whole world is now based on computer programs and therefore there is a large amount of experience out there on how to efficiently run a coding project. We should learn from this experience to ensure the efficient use of funds and to avoid any pitfalls. It might be best to focus on projects in the natural science fields. Computer modelling projects for weather and hurricane forecasting might be specifically relevant and well developed.

Non-scientific software should be considered. The domain for software isn't as important as the success the developers have had in building a pipeline for development/release and collaboration.

What tasks are required to maintain a software package?

There are a variety of tasks that are required to maintain a software package ranging from keeping up to date with compiler versions to logging and correcting bugs. (Provide a comprehensive list).

Who should use the next generation stock assessment model?

The specifications of different aspects of the next generation stock assessment model (e.g., user interface) will depend on the user base. If the user base is anyone, then there should be very strict standards implemented in the user interface to ensure that the model is not used incorrectly. However, if the user base individuals are highly trained and experienced stock assessment scientists, then more flexibility should be given to the user. Given the great need for stock assessments and the limited availability of experts, both scenarios will be necessary and some form of "expert system" is needed to aid the novice in setting up a model, whereas more flexibility is needed for the expert.

Does the next generation model need automatic report writing?

As stock assessment models become more standardised, the potential to develop automatic report writing increases. Having automatic reports would greatly reduce the preparation time and allow analysis to focus on improving the science. However, despite effort to harmonise the way management advice is presented (e.g., the Kobe process) each management agency has its own style. Therefore, the automatic report writing should be modularised and customisable so that automatic reports are easy to produce, but they are flexible.

APPENDIX 2: SURVEY RESPONSES OF DESIRED FEATURES AND FEATURES IN CURRENT STOCK ASSESSMENT MODELS

Two nearly identical surveys were sent to the workshop participants to determine 1) the features desired in the next generation general stock assessment model and 2) the features in current general stock assessment models. The desired features were based on both features used in current stock assessments and those features that assessment authors would like to use but are not available in current models. We received responses from 23 participants for the desired features covering a wide range of species and regions. We received 14 responses from participants for the current features included in general models. The general models are listed in Table A2.1.

Many features were highly desired (see red text in Table A2.2) and several of these were not included in several of the current general models (see highlighted text in Table A2.2). These include major structural features such as area structure, random effects, and mark-recapture data. Currently, no general model includes all three of these features.

Table A2.1. General models and their origins. Those in black are represented in responses to the survey. (Table is continued on the next page.)

Acronym	Name	Organisation	Comment
a4a	Assessment for All	European Commission Joint Research Centre	Part of Fisheries Library in R (FLR)
AMAK	Assessment Method for Alaska	US National Marine Fisheries Service (NMFS) Alaska Fisheries Science Center	Flexible time varying selectivity
ASAP	Age Structured Assessment Program	US National Marine Fisheries Service (NMFS) Northeast Fisheries Science Center	Graphical User Interface
BAM	Beaufort Assessment Model	US National Marine Fisheries Service (NMFS) Southeast Fisheries Science Center (Beaufort)	
CASAL	C++ Algorithmic Stock Assessment Laboratory	National Institute of Water and Atmospheric Research (NIWA), New Zealand	
Casal2	Casal2	National Institute of Water and Atmospheric Research (NIWA), New Zealand	
Coleraine/ Awatea		University of Washington	Bayesian focus
frasyr	Fisheries Research Agency (FRA) calculating sustainable yield (SY) with R	Japan Fisheries Research and Education Agency	VPA
Gadget	Globally Applicable, Area–Disaggregated Ecosystem Toolbox	Marine and Freshwater Research Institute, Iceland	Has multi-species functionality
GMACS	Generalized Modeling for Alaskan Crab Stocks	US National Marine Fisheries Service (NMFS)/University of Washington	Length based
JABBA	Just Another Bayesian Biomass Assessment	Department of Environment, Forestry and Fisheries, South Africa, NOAA-PIFSC	Surplus production model, Bayesian

Acronym	Name	Organisation	Comment
JABBA-Select	JABBA-Select	Department of Environment, Forestry and Fisheries, South Africa, NOAA (multiple)	Adjusts for selectivity
MAS	Metapopulation Assessment System	US National Marine Fisheries Service (NMFS) Pacific Island Fisheries Science Center/South East Fisheries Science Center	Spatial focus, in development
MFCL	MULTIFAN-CL	Secretariat for the Pacific Community (SPC) and Otter Research	Spatial & tagging focus, pioneering
Poseidon		Arizona State University	Agent based
SAIGE	Stock Assessment with Individual Growth Equations	Queensland Department of Agriculture and Fisheries (DAF), Australia	Individual growth
SAM	State-space Assessment Model	National Institute of Aquatic Resources, Technical University of Denmark	Random effects, flexible time varying selectivity
SS	Stock Synthesis	US National Marine Fisheries Service (NMFS) National	Widely used
WHAM	Woods Hole Assessment Model	US National Marine Fisheries Service (NMFS) Northeast Fisheries Science Center	Random effects, covariates

Table A2.2. The proportion of respondents in the desired features and current features surveys that indicated a particular feature was either desired or included, respectively. The highlighted cells indicate where the feature was desired 25 units more than was included in current stock assessment models indicating that this feature would need to be developed for many of the general models before they could be more generally used. The desired features with bold text are those that were desired by more than 50% of the respondents indicating that they are essential features needed in the next generation model. These proportions were arbitrarily chosen simply for illustrative purposes to highlight features. (Table is continued on the next 3 pages.)

Model components	Desired features	Included in current models
Sex structure	**0.65**	0.60
Multi-species technical interactions	0.17	0.13
Multi-species trophic interactions	0.09	0.07
Growth platoons	0.48	0.27
Tagging	0.57	0.40
Area structure	**0.65**	0.33
Stock-structure	**0.52**	0.27
Maximum likelihood	**0.87**	0.87
Bayesian	**0.74**	0.60
Random effects/state-space	**0.57**	0.27
Custom temporal structure	0.48	0.33
Temporal structure varying by fishery	0.43	0.40
Hereditary/Genetics	0.09	0.07
User Interface		
GUI	0.17	0.27
R package	**0.74**	0.73
Comprehensive output processor and display	0.39	0.47
Estimation in phases	**0.78**	0.67
Data		
Catch	**0.91**	0.87
Retained	**0.74**	0.67
Discarded	**0.70**	0.60
Missing / effort-based	0.43	0.33
Units: both biomass and numbers	**0.70**	0.80
Composition data		
Length composition	**0.83**	0.60
Age composition	**0.87**	0.73
Weight composition	**0.52**	0.40
Conditional age-at-length	**0.74**	0.40
Mean weight at age	**0.61**	0.40
Mean length at age	**0.57**	0.27
Sex	**0.74**	0.53
Morph composition	0.13	0.13
Aging error	**0.78**	0.33
Custom composition bin size	**0.52**	0.40
Stock origin	0.17	0.20
Indices		
Index of abundance	**0.96**	0.87
Index of effort	0.48	0.33
Fishing mortality	0.43	0.27
Absolute estimate of abundance	**0.70**	0.67
Environmental index (of a parameter random effect)	**0.61**	0.20
Tagging		
Mark-recapture (to estimate movement and/or F or B)	**0.57**	0.33
Growth increment	0.39	0.20
Close-kin	0.17	0.00
Gene-tagging (Maybe some special requirements)	0.13	0.00
Likelihood functions		
Index		
Lognormal	**0.91**	0.87

Model components	Desired features	Included in current models
Normal	**0.61**	0.40
t-distribution	0.30	0.07
Estimate additive variance	**0.52**	0.33
Estimate multiplicative variance	**0.52**	0.27
Composition		
Modified Chi-squared	0.09	0.00
Multinomial	**0.87**	0.60
Dirichlet-multinomial	**0.61**	0.13
Estimate sample size	**0.52**	0.27
Tail compression	0.30	0.27
Survival analysis (hazard functions)	0.17	0.00
Tagging		
Least squares / robust least squares	0.26	0.00
Poisson	0.26	0.07
Negative binomial / with added zeroes	0.48	0.13
Priors		
Normal	**0.83**	0.47
Beta	**0.70**	0.40
Lognormal	**0.83**	0.60
gamma	**0.70**	0.13
Variance estimation		
Normal approximation	**0.83**	0.53
Profile likelihood (How automatic)	**0.70**	0.13
Bootstrap	**0.61**	0.40
Bayesian		
MCMC		
Metropolis-Hastings	**0.65**	0.40
NUTS	0.43	0.07
Hamiltonian	0.43	0.00
Importance sampling	0.30	0.00
Sub models		
Fishing mortality		
Solve the catch equation (Known catch, hybrid method)	**0.65**	0.33
Pope’s approximation (Known catch)	0.39	0.40
F as parameters and Fit to catch	**0.74**	0.47
VPA	0.17	0.07
Natural mortality		
Age specific	**0.83**	0.73
Size specific	**0.52**	0.20
Time varying	**0.52**	0.60
Sex specific	**0.57**	0.53
Density dependent	0.22	0.13
Growth		
Time varying	**0.57**	0.33
Seasonal	0.39	0.27
Cohort specific	**0.57**	0.27
Growth morphs	0.35	0.20
Density dependent	0.43	0.20
Area specific growth	**0.57**	0.13
Empirical weight-at-age	**0.61**	0.40
Age-specific deviates	0.30	0.13
Growth Model		
Von Bertalanffy	**0.91**	0.60
Richards	0.39	0.13
Spline	0.30	0.00
Growth cessation	0.09	0.13
Recruitment		
Log-normal	**0.83**	0.67

Model components	Desired features	Included in current models
Variable birth dates (influences growth)	0.30	0.13
Area specific stock-recruitment	0.39	0.13
Hermaphroditism	0.22	0.20
Recruitment Model		
Beverton-Holt	**0.83**	0.87
Ricker	**0.52**	0.27
Hockey stick	0.30	0.40
Low-fecundity	0.09	0.07
Maturity / Spawning potential		
Both age and size-dependent	**0.52**	0.47
Flexible length-weight relationship (season, time)	**0.70**	0.47
Selectivity		
Age based	**0.83**	0.73
Length based	**0.78**	0.53
Sex based	**0.57**	0.47
Seasonal	0.43	0.27
Multiple simultaneous selectivities	0.22	0.27
Retention curve	0.39	0.40
Time varying	**0.78**	0.67
Mirroring	0.35	0.40
Selectivity models		
Logistic	**0.91**	0.73
Double normal	**0.78**	0.53
Spline	0.48	0.27
Parameter for each age/nonparametric	**0.65**	0.53
Catchability		
Density dependent	0.48	0.40
Seasonal	0.43	0.20
Function of effort	0.43	0.13
Mirroring	0.22	0.27
Initial conditions		
Fit to equilibrium catch	**0.61**	0.20
Recruitment deviates	**0.70**	0.27
Offset	0.17	0.07
Fishing mortality	**0.65**	0.40
Movement		
Age or size	**0.65**	0.27
Implicit to Non-adjacent areas	0.35	0.20
Movement functions		
Piecewise linear	0.22	0.07
Tagging		
Release/sampling age assignment from length	0.35	0.13
Reporting rate structure	0.48	0.13
Reporting rate priors	0.35	0.13
Reporting rate mirroring	0.13	0.13
Mixing period	0.26	0.13
Temporal variation and Covariates		
Random	0.43	0.27
Random walk	**0.70**	0.47
Time blocks	**0.70**	0.53
Time trends	**0.57**	0.20
Covariates	**0.57**	0.27
As data covariates	0.22	0.13
As structure covariates	0.13	0.07
Multiple covariates, complex relationships	0.22	0.13
Design matrix	0.26	0.20
Diagnostics		
Package for creating stats	**0.78**	0.60

Model components	Desired features	Included in current models
Jittering	**0.70**	0.27
Age structured production model	0.43	0.13
Catch curve diagnostic	0.43	0.20
R0 profile	**0.70**	0.27
Retrospective analysis	**0.83**	0.67
Projections and reference point calculation		
MSY calculations		
MSY	**0.78**	0.87
Spawning biomass per recruit	**0.78**	0.60
Biomass target	**0.65**	0.60
Dynamic B0	**0.57**	0.27
Projections		0.00
Random deviates	**0.52**	0.67
Normal approximation	**0.52**	0.20
Parameter uncertainty	**0.74**	0.53
Fishery impact plots	**0.52**	0.27
Simulations		
Data generation	**0.70**	0.67
Contamination	0.30	0.13
Process error	**0.70**	0.60
MSE (closed loop)	**0.52**	0.40
Project management		
Project management software	0.17	0.40
More than one developer	**0.57**	0.87
Model testing	**0.57**	0.67
Steering committee	0.48	0.20

APPENDIX 3: WORKSHOP ABSTRACTS

A3.1 Coding philosophies, software structure, and underlying language base

Matthew Supernaw: Engineering Practices for Maintainable Software
https://www.youtube.com/watch?v=Y7Hpwvcq710

Software Engineering (SE) provides a systematic and disciplined approach for developing maintainable software. It prescribes a framework, that if followed leads to efficient and highly extensible solutions. Approximately 80 percent of software cost ultimately goes to towards maintenance. This presentation provides an overview of the software engineering process and introduces several key concepts that will help the fisheries community in the pursuit of a generalized stock assessment framework with maintainability and efficiency as a priority from the onset.

Corinne Bassin: Enabling Successful Onboarding of Scientific Tools Via Development Best Practices
https://www.youtube.com/watch?v=EU62mFgnau4

Developing scientific software applications for operational usage across scientific and policy organizations can be challenging. The variety of both development processes and user interface preferences for operational tools can create impairments to onboarding the applications for general usage. When considering new tools for operational use it is important to consider strategies to ensure ease of collaboration on tools, ensuring reproducibility of tools and results, scientific integrity, and simplicity of user interfaces. These basic tenants can more easily be ensured by implementing practices from general software development that may be used by developers early in the coding process. As part of the NOAA Fisheries Integrated Toolbox, we are focusing on strategies to help developers think ahead of time about using software development best practices which will make onboarding tools simpler and allow for greater usability and continued development of these tools into the future.

Scott Rasmussen: Casal2 – Dynamic Modelling Framework
(no video available)

This is a talk about the Casal2 modelling platform in development led by NIWA. This is a highly dynamic and scalable modelling platform that allows a wide range of different modelling techniques and styles.

Ernesto Jardim, Finlay Scott, Paris Vasilakopoulos, Cecilia Pinto, Alessandro Mannini, Christoph Konrad, Iago Mosqueira: A modular framework for the generic application of fisheries management strategy evaluation.
https://www.youtube.com/watch?v=4Xi1YwFCNHg

Management strategy evaluation (MSE) is a complex simulation and forecasting procedure that takes into account structural and observational uncertainty of both stock dynamics (growth, recruitment, maturity) and exploitation by fishing fleets (selectivity, effort). The MSE paradigm leads to the articulation of a decision-making framework for fisheries management under uncertainty. Within the ‘assessment for all’ (a4a) initiative, the European Commission’s Joint Research Centre has developed a modular MSE, built with the R programming language and implemented in the Fisheries Library in R (FLR) toolbox. The a4a MSE algorithm includes the most common elements of uncertainty and allows, among others, the formulation of alternative management procedures and harvest control rules, testing the robustness of reference points, etc, within a reasonable operational time frame. The a4a MSE has a modular design, meaning that the system components are divided into smaller, independent, parts (modules). These modules link back to the MSE model parts, so that each element of the model refers to a single module. Here, we present the a4a MSE algorithm and describe its application to a range of case studies. These applications illustrate the flexibility of the a4a MSE algorithm and its potential to support the evaluation of multi-annual management plans, identification and testing of management procedures for data poor stocks, development of harvest control rules, etc.

Wen-His Yang: Doing Stock Assessment Using ADMB, Stock Synthesis or TMB? A Case Study on the Queensland Saucer Scallop
https://www.youtube.com/watch?v=syCxROUmwMc

The Queensland Government manages fisheries to maintain their sustainability, relying heavily on the results of stock assessments. For future stock assessments, the Queensland Department of Agriculture and Fisheries (DAF) aims to improve the stock assessment process in terms of accessibility, repeatability, and swiftness. To address this, DAF organised a Stock Synthesis workshop in August 2019 in Nambour, Queensland with participants consisting of DAF scientists, and mathematicians and statisticians from the Centre for Applications in Natural Resource Mathematics (CARM) at the University of Queensland. The Stock Synthesis package differs from ADMB and TMB which are C++ based languages for optimisation with more flexibility for model development. ADMB has been used for stock assessments of species in the Queensland waters. In this talk, we present the assessment of the Queensland saucer scallop (*Ylistrum balloti*) stock using ADMB, Stock Synthesis and TMB, sharing our experiences in using the three programs for the scallop assessment.

A3.2 Stock assessment model features

Momoko Ichinokawa, Shota Nishijima, and Hiroshi Okamura: Stock assessment model in Japan: past to present
https://www.youtube.com/watch?v=4yOQWIBxhWY

The stock assessment method used in Japanese fisheries stocks have been evolving independently from international standards (such as statistical integrated assessment models). Tuned-VPA with using a spreadsheet application with graphical user interface (such as Microsoft Excel ®) has the long history of the Japanese stock assessment model, being the current majority especially for data-rich and TAC-managed stocks since the introduction of TAC management system of 1998.

This presentation reviews our efforts to break away from Excel to R since 2012 in the stock assessment of Japanese fisheries. Our developed R programs can conduct tuned VPA, calculate biological and empirical reference points, and conduct stochastic future projections. The development of R programs is now fully achieved through some important breakpoints such as complete replication of the past stock assessments by using Excel for TAC species (Ichinokawa and Okamura 2014, Bulletin of Japanese Society of Fisheries Oceanography), incorporation of “ridge-VPA” (Okamura et al 2017, ICES Journal)”, calculation of MSY-based reference points under stochastic simulations with hockey-stick stock-recruitment relationship (Ichinokawa et al 2017, ICES Journal), and shortening the calculation time with TMB (Nishijima, unpubl.). The R programs are (not-officially) available as a R package named by “frasyr” (FRA-sustainable-yield-with-R) in github website.

In conjunction with new Japanese policy on fisheries management toward explicit MSY-based management, “frasyr” will be used as the main software in the future Japanese stock assessment. There are, however, still many operational problems on this R package: “generality”, “usability”, “co-developing”, “authorship”, “efficiency”, and “transparency”. We listed up our current facing problems in this presentation and pass those issues to the next generation’s scientist who can blueprint the framework of next generation stock assessment software in Japan.

Akira Hayashi, Junji Kinoshita, Akihiro Manabe: Stock assessment model in Japan: future perspective
https://www.youtube.com/watch?v=qWroF4Bp6d8

Assuming the next generation stock assessment models are built how should we utilize it for the actual stock assessment? This presentation will address the desired direction inspired from a concept of "DevOps": a cultural movement in the field of software business, which enables producing products efficiently in sustainable manner.

The global standard of stock assessment has been improving in terms of development procedure the program for stock calculation; codes are version controlled with automated tests, and programs are distributed as stable package. But the development of stable programs is not the goal of the stock assessment, but only the beginning to achieve our ultimate goal: sustainable fishery with consensus among stakeholders. Until then, researchers routinely have to estimate stock status with programs, write reports, communicate with stakeholders, re-calculate to cope with new requirements, and even write additional reports.

In Japan, the work surrounding stock assessment has various problems in the operational aspect.

Although the component program (frasyr: introduced in the previous presentation) is distributed as a stable package, the current workflow remains vulnerable and slow because of (1) the user-end script which have to be run manually, (2) researcher works on the similar scenarios concurrently with other researchers on different stocks, and (3) errors that may occur in a recurrence manner. To make consensus among stakeholders on stock management, scenarios of stock calculation should be more meticulous, and the number of review-recalculate cycles should be increased. We will encounter too much burden if we lack the operational framework for utilization of the stock assessment software.

In the world of software business, cycles of development and operations have become harmonized owing to the permeation of "DevOps". The key of their success was rooted in the idea that the most of "Ops" procedures such as running program and making document should be automated and integrated into "Dev" part. In the world of stock assessment, aiming sustainable fishing, how does our "actual" work consist of? Essentially, we have to:

- solve a problem
- calculate outcome from data
- ensure reproducibility of the calculation
- update calculation to meet new requirements

This presentation will translate and link ideas from DevOps into our field and suggest the operational framework for the next generation stock assessment which allows us to focus on improving customer satisfaction: sustainable fishing with full consensus.

Kota Sawada: Case studies of the local stock assessment in the Northwest Pacific: difficulties in the stock assessment for seamount bottom fisheries
https://www.youtube.com/watch?v=bZMtwSTPfPk

Although the stock assessment is crucial for the sustainable use of biological resources, it is often hindered by various kinds of difficulties. In this presentation, I review the current status of stock assessment/management for bottom fisheries in the Emperor Seamounts area which offers a challenging area of development in the stock assessment models incorporating various difficulties such as life history diversity and limited data quality. I explain why the stock assessment for bottom fisheries resources in the Emperor Seamounts area, especially North Pacific armorhead *Pentaceros wheeleri*, has not been successful despite the long effort by North Pacific Fisheries Commission (NPFC) and its predecessor.

North Pacific armorhead has a unique life history, consists of long pelagic period, determinate growth and body weight reduction (Kiyota et al. 2016). Length-based methods are inapplicable because they do not grow after recruitment. Catch and stock levels are highly unstable due to recruitment fluctuation and high fishing pressure, and thus difficult to predict by population dynamics models. Unreliability in fine scale catch data further complicates the analysis. Catch per unit effort may be hyperstable because of aggregative behavior and affected by target shift between primary and alternative stocks. Target shift is more important in CPUE of the alternative target (splendid alfonsino *Beryx splendens*) rather than that of the primary target (armorhead), as indicated by “directed CPUE” analysis (Biseau 1998).

Several approaches, including De Lury method, surplus production model and yield-per-recruit analysis, has been adopted to assess North Pacific armorhead and splendid alfonsino in the Emperor Seamounts area. Although those approaches succeeded partly, the stock management based on solid assessment has not been implemented by NPFC. Instead, to protect spawning stock and juvenile individuals, flexible catch limitation depending recruitment level and mesh-size regulation started this year, under the name of “adaptive management”, based on the limited scientific knowledges and no quantitative evaluation of stock status. Finding a way to deal with such data-poor and atypical stocks and to reconcile with adaptive approach will be an important task for future studies on stock assessments.

Shin Fukui: Case studies of the local stock assessment in the Northwest Pacific: application of robust regression in estimating stock-recruitment relationship (15)
https://www.youtube.com/watch?v=Q8PAcrXT-98

Fisheries stock assessment data sometimes have strong year classes of recruitment that can often give a significant impact on population dynamics of fisheries resources. The strong year class can also affect the estimation of stock-recruitment relationship when using Least Mean Squares (LMS) method with a log-normal distribution of recruitment deviations if the year classes are unusual strong like as outliers. In such a case, the stock assessment using LMS becomes biased due to the unusual strong year class.

To avoid stock-recruitment relationship being sensitive to extraordinarily strong year classes, it could be preferred to estimate SR relationship by reducing the weight on the information of outstanding recruitment. Least Absolute Deviation (LAD) method is the parameter estimation method by minimizing the sum of absolute values of deviations. LAD assumes a Laplace distribution as the error structure and has been used as the common method for robust regression even when extreme outliers exist in data.

In this talk, we show some simulation results to demonstrate usability of LAD compared with LMS when there are outliers of recruitment deviations. Also, we introduce case studies in Japanese fisheries stocks when LAD method is applied to estimate SR relationship in the stock assessment.

Jemery Day: Stock assessment issues in South Eastern Australia
https://www.youtube.com/watch?v=Ok7X_zN1i_Q

Australia's Southern and Eastern Scalefish and Shark Fishery (SESSF) is a multi-sector, multi-species fishery covering almost half of the Australian Fishing Zone, managed by limiting catch (TAC), restricting the number of boats and regulating gear. There are 34 different species managed under a quota system, with around 13 Tier 1 (data rich) assessments. These were all assessed using bespoke models up until 2006, when most assessments were transferred to Stock Synthesis (SS). With assistance and training from the stock synthesis team at NOAA, we now have 13 stocks assessed in SS (all non-shark Tier 1 stocks). These assessments range from some which are relatively straightforward, to others using little used features in SS, including cohort dependent growth, productivity shifts and projections with low recruitment scenarios. To provide management advice, the Australian Harvest Control Rules were coded within SS so that Recommended Biological Catches (RBCs) are explicitly calculated within the software. Incorporating Australian Harvest Control Rules within the software was essential to allow us to adopt this package and is an important requirement for future general stock assessment models. There are advantages with a common framework, through consistent presentation of assessments across species to both managers and industry (due to the r4ss output package), and a much easier process of review, collegial work within our team and transferring assessments between different assessment scientists. In contrast to this move towards adopting a common package in the SESSF, the Macquarie Island Patagonian Toothfish assessment has recently moved from an assessment using a modified version of SS (a spatial model with tagging data where the age and sex at first tagging is unknown), into TMB to allow the tagging component to be modelled more accurately and more efficiently and to potentially allow the use of close kin mark-recapture in future. General stock assessment models have been useful in Australia, but they currently do not satisfy all objectives required from such tools.

André E. Punt: Essential Features of the Next-Gen Integrated Assessment
https://www.youtube.com/watch?v=h_ifBwy5d7E

Integrated (or statistical/ stage-space) methods have been the preferred approach for conducting stock assessments and providing the basis for management advice for fish and invertebrate stocks off South Africa, Australia, New Zealand and the west coast of North America since the publication of seminal paper by Fournier and Archibald in 1982, and now most assessments worldwide have adopted this approach. Methods to assess stocks based on single-species, single-area age-structured models are now standard, with the major debates associated with these models related to data choice and data weighting. The major challenges for next-gen stock assessments are the extensions needed to assess stocks that do not satisfy the 'single-stock fish' paradigm, and sadly it is increasingly becoming obvious that many stocks fall into this category. My talk will highlight the following areas: (a) spatial models, and in particular modelling multiple populations (e.g., stocks that exhibit a cline in density and population structure over their range), (b) models that are able to capture the age and length dynamics of populations simultaneously yet computationally efficiently, and (c) models that can scale from data-rich to data poor (and include as special cases the many bespoke approaches to data-poor assessment). In relation to data, there is a need to ensure that the next-gen stock assessment methods better handle tagging data (age-length models may help in this regard), in particular to be able to use genetic mark-recapture data (next-gen-x-2 stock assessments may be built around such data), and to assess multiple stocks simultaneously (satisfying the promise of 'Robin Hood' paradigm). Some challenges that have plagued stock assessment for decades warrant continued attention (at the theoretical and applied level) such automatic data weighting and tuning, improved coding to facilitate application of state-of-the-art methods for quantifying uncertainty, and adoption of true state-space formulations to allow more parameters to be treated as random effects.

Erik H. Williams and Kyle W. Shertzer: Beaufort Assessment Model (BAM): Lessons Learned From Twenty Years of Software Development
https://www.youtube.com/watch?v=Sv_DtjWqguw

The Beaufort Assessment Model (BAM) was developed out of necessity during the early years of AD Model Builder in 2001. Named for the NOAA Beaufort Lab in North Carolina, USA, BAM was developed for stock assessment analyses of fish species within the South Atlantic Fisheries Management Council jurisdiction and coincided with the beginning of the SouthEast Data, Assessment and Review (SEDAR) process. These developments coincided with a shift in this region from VPA models to forward projecting statistical catch-at-age models (now called, integrated models). The BAM system went through much evolution to reach its current form. Currently it consists of code and packages written in R, ADMB, C++, and LaTex for reading in data, running assessment models and forecasts, processing output, and generating reports. Feedback from long time and new users of the BAM system have provided insights into important features for a successful next generation stock assessment system.

Allan C. Hicks, Ian J. Stewart, Piera Carpi, David T. Wilson, Steve Berukoff: Needs from a next generation general modelling framework to support the future of stock assessment and MSE at IPHC
https://www.youtube.com/watch?v=OXr2N7MNSIo

The International Pacific Halibut Commission (IPHC) manages the Pacific halibut (*Hippoglossus stenolepis*) resource for the governments of Canada and the United States of America, and a major responsibility is providing scientific support for setting annual catch limits. Advice from the annual stock assessment is provided by an ensemble of four stock synthesis (SS) models presented in the form of a decision table reporting the risks to (probabilities) of stock and fishery performance metrics given projected catch levels. Results can be integrated via MCMC samples or Monte Carlo generation of distributions with MLE, but to properly integrate MLE results, the variances and covariances of specific quantities are needed, some of which are not available in SS (e.g., dynamic biomass reference points).

A Management Strategy Evaluation (MSE) is underway at IPHC with the development of a simulation framework and spatially-explicit operating model for Pacific halibut. An initial framework used SS as an operating model by writing R code as a wrapper to condition the OM and perform closed-loop simulations to compare among management procedures. Other agencies have performed MSE simulations using SS (e.g., Pacific hake), and many others are also interested in a general framework to conduct MSE. Challenges in transforming a generalized stock assessment model into an MSE framework include: 1) conditioning the OM, 2) producing multiple starting points from which to simulate future trajectories, 3) incorporating a management procedure in a closed-loop simulation, 4) using the OM to simulate into the future with variability on both parameters and processes, 5) access to population quantities to simulate observations, and 6) outputs that can be transformed into performance metrics.
In the future IPHC needs a general modelling framework that can 1) be used to develop multiple structurally differing models with consistent outputs for use in both an ensemble and computationally efficient sensitivity analyses, 2) incorporate a range of modern fisheries stock assessment techniques and provide options to compare between them (e.g., data weighting, functional forms), 3) output dynamic reference points (time-series and equilibrium), 4) provide variances and covariances of both parameters and derived quantities, 5) provide options for various time-efficient MCMC algorithms, 6) be used in a simulation framework to rapidly examine estimation performance, explore alternative hypotheses about dynamics and observation processes, and perform closed-loop simulations for use in MSE., 7) allow for expansion and incorporation of new data sources, modelling assumptions, parameters, and outputs, 8) provide standardized outputs that foster collaboration between agencies and efficient review processes, and 9) easily link with software to visualize and summarize results (e.g., r4ss). We discuss how IPHC has recently utilized general stock assessment software, some of the challenges experienced, and specific needs for a future generalized modelling platform.

Divya Varkey, Jonathan Babyn , Paul Regular, Rajeev Kumar: Hybrid – a modelling framework to sidestep structural uncertainty in models
https://www.youtube.com/watch?v=k0EDMxQCBsM.

The fisheries stock assessment modelling community has made a lot of headway in ways to present and evaluate different types of uncertainties in state-space stock-assessment models. Yet, we have limited opportunity to evaluate structural uncertainty stemming from model structure and parameterization. In recent years, there are several papers where authors have presented and compared alternate models for assessment of a stock with a. different parameterizations of selectivity, natural mortality, and catchability (Rossi et al., 2019); b. different approaches to model natural mortality (Miller and Hyun 2017); c. likelihood choice (Albertsen et al. 2016) and others. Yet, often in these comparisons, the structure of the base model remains unchanged (Rossi et al 2019) (not always—Miller and Hyun compare SCAA and state-space models). We present 'HYBRID', a modelling framework which allows different choices about the underlying model structure wherein the differently structured models (like Nielsen and Berg 2014; Cadigan 2015) are realizations within the same modelling framework. Features can be turned an off depending on user choice. The first and most prominent is different options for modelling F: a logistic flat topped or dome selectivity with year effects, a correlated random-walk, or a correlated auto-regressive structure over age or year; additionally it is possible to break the processes for F in years where important events (like a moratorium) might have happened in the fishery. Other choices built into the model are (i) alternate parameterization of M, (ii) choice of modelling recruitment, (iii) choice of using censored likelihoods for missing data points, (iv) choice of fitting to catch numbers at age or fitting to catch proportions and landings, (v) censored fitting of landings data when there is uncertainty about landings in different time periods in the landings, (viii) year effects in survey where needed, and (ix) use of correlated likelihoods in fitting catch-at-age or index-at-age. Further, a flexdashboard mechanism allows comparison of residual patterns and stock status between models. We expect that this modelling framework can support researchers in fisheries stock assessment when testing out new models for a stock especially in new model evaluations like 'Benchmark' processes at ICES; 'Assessment frameworks' in Canada; and 'SEDAR' in the US, and other similar endeavours elsewhere.

Albertsen, C. M., Nielsen, A., & Thygesen, U. H. (2016). Choosing the observational likelihood in state-space stock assessment models. Canadian Journal of Fisheries and Aquatic Sciences, 74(5), 779-789.
Cadigan, N. G. (2015). A state-space stock assessment model for northern cod, including under-reported catches and variable natural mortality rates. Canadian Journal of Fisheries and Aquatic Sciences, 73(2), 296-308.
Miller, T. J., & Hyun, S. Y. (2017). Evaluating evidence for alternative natural mortality and process error assumptions using a state-space, age-structured assessment model. Canadian Journal of Fisheries and Aquatic Sciences, 75(5), 691-703.
Nielsen, A., & Berg, C. W. (2014). Estimation of time-varying selectivity in stock assessments using state-space models. Fisheries Research, 158, 96-101.
Rossi, S. P., Cox, S. P., Benoît, H. P., & Swain, D. P. (2019). Inferring fisheries stock status from competing hypotheses. Fisheries Research, 216, 155-166.

Chantel Wetzel, Jason Cope: Moving up the assessment ladder: A flexible and integrated approach to modelling data-limited stock assessments
https://www.youtube.com/watch?v=7dToRrdkTq0

Approaches for scientifically informing management of data-limited stocks are numerous and continuously expanding, highlighting the diversity of data availability and global need for analytical options. These methods typically use life history inputs and limited information to calculate harvest levels, fishing rates, and/or relative stock status, and often share similar, numerous and restrictive assumptions. The number of methods can seem overwhelming, and moving from platform to platform depending on the available data increases complexity and chances for mistakes. Constructing a single integrated stock assessment framework that traverses data-limited to data-rich options is an appealing alternative. Stock Synthesis, a statistically integrated age-structured model, is the most commonly used tool to estimate population size and status for U.S. West Coast groundfish. Here we demonstrate the Stock Synthesis framework as applied to a variety of data-limited approaches. These methods harness the benefits of applying an age-structured population dynamics model with flexible parameterizations to explore uncertainty in life history parameters and selectivity and avoiding certain simplifying assumptions common to many data-limited methods. Applying a single modelling platform can facilitate transparency and comparison of model results across the data spectrum by providing a clear way of including additional data when it is available, thus moving up the assessment ladder.

A3.3 Groups / partitions

Nick Davies, Dave Fournier, John Hampton: Partitioning in MULTIFAN-CL in respect of space, tagged populations, species, stocks, and gender – coding implementation and recommendations
https://www.youtube.com/watch?v=d5PBDCYLczw

MULTIFAN-CL is an integrated, statistical, age-structured, length-based model that has routinely been used for stock assessments of tuna and other pelagic species since the 1990's. It is typically fitted to total catch, relative abundance, size-frequency and tagging data stratified by fishery, region and time period. The implementation of partitions for gender, space, tagging, and species or stocks in MULTIFAN-CL is presented, offering an example for generic fisheries software. The spatial partition is a core data structure, being the primary dimension of the population state matrix: by region, time period, and age class. Movement among regions may be parameterized with independent transfer coefficients or with orthogonal-polynomial coefficients, that may be constant or age-specific. Similarly, the tagged partition is implemented as a "parallel" core data structure having four dimensions: release event, region, time period, and age class. While separate from the untagged partition in respect of recruitment (being the initial tag release samples), it shares the fundamental processes and parameters for natural and fisheries mortality, movement, and growth. Partitions for species, stocks and gender were recently added by means of an adaptation of the spatial partition in respect of regions, to create "mirrored" regions for the additional partitions.

Code development began in 2011 and was largely completed by 2014. Initial development was in respect of species, that was later adapted for stocks and sexes being special cases of “species”. While the simultaneous implementation of the species-sex, or stocks-sex partitions is technically possible, this has not yet been attempted with real examples. The creation of mirrored regions for additional species is described, and, also the processes and parameters that may be mirrored, shared, or estimated as unique in each region corresponding to each species/stock/sex. The regional structure for these partitions readily allows for the partitioning of tagged populations among species/stocks/sexes, because tagging events are themselves partitioned in respect of regions. Data may be aggregated among each species/stock/sex. Tagging events are mirrored in the regions for the additional species, and the proportions assigned to each species is assumed to be as estimated for the untagged population, thus creating multiple tagged populations, one for each species/stock/sex, from the single release event. Similarly, apportioning aggregated tag release groups among sexes must take account of the sex-specific growth rates and sex ratios. Examples for each case of multiple species/stocks/sexes are presented to demonstrate the implementation of the partitions.

Coding considerations when developing new partitions within existing code are discussed, including validation methods. For example, it needs to be ensured that a multi-species model is capable of reproducing the results of the two single-species models fitted to each species' data separately. Ensuring the code’s integrity throughout the development process is critical, and the validation procedures used for MULTIFAN-CL are described. This aims to prevent new features being developed for a single species case “breaking” the code when used in the context of a multi-species/stock/sex model. Benchmark testing therefore includes a range of single- and multi-species/sexes test data sets. Based on our experience, recommendations are offered for future developers in respect of coding a multi-partitioned generic model, perhaps beginning with the first rule: establish the correct data structures at the outset.

Richard McGarvey, Richard D. Methot, Ian G. Taylor, André E. Punt, John E. Feenstra, Janet M. Matthews: Will age-and-length based modelling permit broader application of the next-generation fishery assessment model?
https://www.youtube.com/watch?v=T2MZx3Iln3c

One of the questions for a next-gen model is whether to make population numbers dynamically dependent on both age and length-within-age. A second question is whether such an age-and-length based formalism can be applied to both age-based assessments (where otoliths are read) and to length-only assessments. If so, only a single set of software would need to be developed and supported. We compare two current methods for age-and-length-based fishery models (platoons in Stock Synthesis and slice partition). Simulated data from an individual-based model that is independent of these two approaches will be generated for model testing. We also discuss the relative advantages and disadvantages of the two methods.

James Ianelli, Kirstin Holsman, James T. Thorson: Assessment developments including climate enhanced multi-species models from the North Pacific.
https://www.youtube.com/watch?v=H3x05da72sE

We note that flexible mechanistic models linking physical and biological dynamics to fisheries problems continue to be developed and link to the future climate change scenarios. In the North Pacific, part of this effort includes scaling simple single- and multi-species within these dynamics. We start with a survey of innovative developments within single species models to better account for observed changes in species distribution and treatment of survey data. We expand some of these approaches to trophically linked key species with an example from the eastern Bering Sea. Options to feed time and age-varying predation mortality into simpler single species models are presented and discussed. Finally, we project under alternative climate scenarios given linked spatio-temporal predictions of environmental conditions. The attributes and trade-offs of integrating more complicated models into clear tactical and strategic management advice will require transparency to be effective.

Katyana A. Vert-pre, Nicholas Payette, Ernesto Carrella, Michael Drexler, Jens Koed Madsen, Steven E. Saul, Richard M. Bailey, Aarthi Ananthanarayanan, Chris Dorsett: Are Agent-Based Approaches the Future of Fishery Management? – applying the Poseidon model to the Eastern Pacific Tropical Tuna Fishery.
https://www.youtube.com/watch?v=rV7Hsjiqwpg

Fisheries are complex, adaptive, coupled social-ecological systems. The behavioral and social responses of individual and groups of fishers to fisheries management interventions remain a critical knowledge gap in developing robust management solutions. Management and implementation uncertainty can dramatically undermine an adaptive management paradigm. Therefore, research to understand and forecast human systems are needed, but hypothesis-driven, controlled experiments are difficult or impossible to conduct in complex systems like fisheries. Agent-based models (ABMs) are well suited to address these challenges in the human system as they capture the motives of individual actors in the system but emphasize the emergent, group level, properties which are relevant to management. Furthermore, fully coupled social-ecological models can extract additional information about the fishery system that may not be captured from either the biological, social, or economic dimensions individually. An implementation of an agent-based model framework, “POSEIDON” is in development. It allows to evaluate management alternatives when using fish aggregating devices (FADs) in the Eastern Pacific. The coupled social-ecological model represents the biologies of three tropical tuna species and draws the dynamics of purse seine vessels and fish aggregating devices using adaptive agents. Once fully developed and calibrated, this model can be used to evaluate the trade-offs associated with management alternatives for the FAD fishery across a spectrum of biological, ecological, and economic criteria. While powerful, Poseidon requires expertise in multiple fields between project managers, social scientists, computer scientist, and fishery scientists. We explore the strengths and challenges brought by working with such a diverse group of scientists on a complex systems model.

A3.4 Observation models

John Feenstra, André E. Punt, Richard McGarvey: A framework for multi-year Leslie-Davis depletion modelling and its use as a stock assessment model feature.
https://www.youtube.com/watch?v=R8W07buyxDo

An overview is provided of the multi-year Leslie-Davis depletion modelling framework of Feenstra et al. (2017), referred to as "EDM", followed by representations of it within integrated modelling. EDM estimates yearly total recruitment numbers, as well as exploitable population at the start of each time step within a part of the fishing season ("fitted depletion period" or "FDP"). Data requirements of EDM consists of total fishery catch in number for all of the season, but catch rates for only the FDP, which may be much shorter than the entire fishing season. The basic assumptions of the Leslie-Davis model of constant catchability and no recruitment apply for the FDP for each season. EDM hence does not require data on total fishing effort during a season, nor does it make assumptions about catchability during the non-FDP. The basic ("base") EDM version estimates a parameter for the population in the initial fishing season, a recruitment parameter for each season, and a single catchability parameter shared among all seasons. It may also be used to estimate exploitable biomass (instead of population numbers) for fisheries that only have catch rate data and catches in weight, but the recruitment parameters in this case estimate total positive production similar to the "production function" in surplus production models. EDM here is applied to the Southern Zone rock lobster fishery of South Australia, as per Feenstra et al. (2017) but the time series of data is extended by four seasons ending in calendar year 2018. Sensitivity analyses comparing standalone EDM results with those from standalone runs of a length-based and catch-conditioned integrated stock assessment model used for rock lobster in South Australia (“LenMod”) reveal that selectivity configuration in the latter determine strongly the level of congruence for exploitable abundance (but not recruitment to legal size). Adding EDM functionality as a feature to a general stock assessment model either by providing it as a standalone option, or as an influence when fitting an integrated model to diverse data. Requirements for integration of EDM within a more general model include that it allows: catch-conditioning, homogeneous catchability over specified contiguous time steps within a fishing season during which substantial depletion occurs and catch rate is fit (i.e., FDP), recruitment to occur outside of the FDP, weighting values for likelihood components. Fitting to length composition data may complement EDM by better modelling of exploitable abundance through informing on selectivity, the latter implicitly assumed to be uniform in standalone EDM. Similarly, setting catchability to be homogeneous for some time steps (when deemed realistic) in integrated models alters estimates and configuration of length selectivity. EDM functionality may be beneficial in data-limited (or -moderate) fisheries to estimate abundance/biomass, and in more data-rich fisheries by enabling the possibility of detecting yearly changes in catchability or by providing an alternative set of abundance/biomass estimates.

Feenstra, J., Punt, A.E., McGarvey, R., 2017. Inferring absolute recruitment and legal-size population numbers of southern rock lobster (*Jasus edwardsii*) in South Australia's Southern Zone fishery using extended forms of depletion modelling. Fish. Res. 191, 164–178.

J. P. Eveson, R. M. Hillary, T. A. Patterson: Integrating conventional and electronic tagging data into the next generation of stock assessment models
https://www.youtube.com/watch?v=sLb7r-1T02s

The job of assessing the status of a given fish stock is often hard enough for a single spatial area. When one cannot really ignore the spatial distribution and/or migratory dynamics of the population being assessed it only gets harder. The most common information deficit is data that can meaningfully inform on spatial dynamics, and mark-recapture data have always been the most likely general data stream that could solve this problem, though they present significant challenges when modelling the full suite of release and recapture data. Several stock assessment packages already have tagging data modules within their general structure, but not for the kinds of electronic tagging data platforms (e.g., archival, satellite, acoustic) that have been increasingly developed and implemented for a variety of species – particularly for pelagic species like tunas, billfish, and sharks. This talk uses two examples (southern bluefun tuna and striped marlin in the South Pacific) to demonstrate how we can go about integrating conventional recapture data only (not releases and recaptures), and the current suite of possible electronic tagging data for the purposes of movement estimation within integrated stock assessment models.

Hans J. Skaug: Towards Close-Kin Mark-Recapture software
https://www.youtube.com/watch?v=PwIlr1OQ9MQ

Mark-recapture methods are commonly used to study wildlife populations. Taking advantage of modern genetics one can generalize from "recapture of an individual" to "recapture of a closely-related kin", and thereby the name Close-Kin Mark-Recapture (CKMR). The basic logic is: if your sample contains many close relatives (genetically determined) then your population is small, and vice versa. The method has been successfully applied to some fish populations, and there is also interest in applying CKMR to terrestrial populations. I will explain basic principles of CKMR and discuss to which extent it is possible to develop generic CKMR software. Part of the conclusion will be that every target organism have their own peculiar life history, which must be accounted for when calculating "recapture probabilities", and this makes it hard to write generic software.

Robin Thomson, Mark Bravington and Rich Hillary: Application of Close-Kin Mark-Recapture
https://www.youtube.com/watch?v=zQFitTcp6uw

Close-Kin Mark-Recapture (CKMR) is an exciting new method that can provide precise time-series estimates of absolute abundance, as well as independent estimates of mortality rate (fishing and/or natural), relative-fecundity-at-size, and population age structure. There is no need for fishing effort or CPUE data, and no need to worry about time-varying catchability or selectivity.

Operationally, CKMR entails collecting many tissue samples together with associated length and (at least some) age measurements, followed by genetic sequencing and the identification of close relative pairs, specifically parent-offspring and half-sibling. Although sample size requirements can be high, costs can be kept low because it is generally sufficient simply to sample from the catch, since the method is not susceptible to vagaries of fishery behaviour. A model that uses CKMR data needs little else besides a catch-at-length time series.

CKMR estimation uses a standard age-structured population dynamics model, combined with special-purpose log-likelihood code that has a clear logical structure. While there is real value to running purely-CKMR models, eg to check the reliability of conventional but suspect data such as CPUE, it is also straightforward in principle to embed the CKMR log-likelihood alongside other data within a classical age-structured integrated stock assessment model framework; at CSIRO, we have used both approaches in the course of addressing Southern Bluefin Tuna, school shark, and several endangered sharks. We present some principles for how this exceptionally powerful technique should--- and should not--- be incorporated into stock assessment software.

Mark Chambers: Implications of entrainment for fisheries stock assessment
https://www.youtube.com/watch?v=kNGJOz8AMY8

There are many examples of the collapse of fisheries exhibiting population dynamics that contrast sharply with stock assessment models. Petitgas et al. (2010) argue collapsed populations, distinct from depleted ones, have suffered disruption to population memory of traditional migratory routes. The social transmission of traditional migratory routes, or “entrainment”, has been taken most seriously for Atlantic herring whose seasonal migrations have been documented for centuries. However, attempts to simulate entrainment dynamics are yet to demonstrate its “survival value”. I show using a simulation loosely based on juvenile southern bluefin tuna that entrainment can lead to increased fitness over fixed probability migration choices in the usual case when mortality varies over space. I also demonstrate how entrainment can contribute to fishery collapse and delayed recovery.

Jonathan Babyn: A New Approach to Generating Spatial Age-Length Keys. Based on Using a Gaussian Field Approximation with Support for Physical Barriers
https://www.youtube.com/watch?v=qAxUoSDf7r4

Estimating the age composition of a fish population is an important first step in the stock assessment process. Often this is done through the use of an Age Length Key (ALK), which links a subsample of fish that have had their ages determined to those that have only had their lengths measured in order to obtain an estimate of the age structure of the entire sample. ALKs can suffer from data gaps and sampling artefacts and are limited in how the ways spatial information can be incorporated. We propose a novel spatial ALK model that uses an approximation of a Gaussian Field and has the ability to account for physical barriers (e.g., coastlines) in the study area. Our approach is compared with a previously suggested spatial ALK model as well as non-spatial approaches using both real and simulated survey data. We find that using a spatial ALK approach can reduce the error for stratified estimates of abundance at age.

A3.5 Management quantities

Richard D. Methot: Stock Synthesis Completes the Cycle: Assessment – Management Quantities – Projections – MSE.
https://www.youtube.com/watch?v=yuFQAxpuIQY

The Stock Synthesis (SS) assessment approach links age-structured assessment of a stock’s exploitation history, with calculation of reference points and then with projections. Movement, seasonality, growth, fleet selectivities, etc. are estimated from maximum likelihood fits to data in the time series, then applied equivalently in equilibrium and projection calculations. A target *F*, say F35% or Fmsy, calculated in the reference point phase can be used as the *F* for the projections. This approach preserves parameter gradient information through these linked modules and propagates variance of parameters estimated for the time series onto the equilibrium calculations and projections. The capabilities of the projection module are extensive to deal with real-world fishery management situations such as uncontrolled bycatch by other fisheries, fixed allocations between sets of fleets, caps on total catch, etc. A novel aspect of projections with SS is its three phased approach. The first phase calculates future catch if fishing was according to Fmsy; the second phase applies fixed input catches, catch constraints and allocations, and biomass-linked control rules including a buffer below Fmsy. Catches from the second phase are save as if they are a time series of future fixed catch quotas. Then the third phase uses these saved catches to calculate future realized F's taking into account stochastic recruitment and implementation error. New work is wrapping the SS framework within a R-based system to conduct Management Strategy Evaluation.

Nathan Vaughan, Kathryn Doering, John Walter, Rick Methot, Skyler Sagarese, Matthew Smith, Nancie Cummings, and Nick Farmer: Management strategy evaluation made operational with Stock Synthesis https://www.youtube.com/watch?v=_N8JeySopzY

Management Strategy Evaluation (MSE) is designed to holistically evaluate alternative management strategies, data collection approaches, and modelling structures. It iteratively simulates the stock assessment, management, and population & fishery processes to evaluate uncertainty and determine which components could be improved to achieve performance. While MSE is becoming routine, the task of conducting it is currently limited by the challenge of creating realistic operating models (OMs) for the population and fishery processes. The stock assessment software package Stock Synthesis (SS) currently represents one of the most complete, generalizable population assessment models available, which is why it is used for a large number of stock assessments around the world. The structure of SS models provides the necessary architecture for MSE OMs, while the stock assessment process formalizes the parameterizing, fitting, and evaluating of models. Pre-existing stock synthesis assessment models therefore represent a potential pool of excellent operating models. Attempts that have been made to utilize this existing resource for MSE are often simplified versions of the original SS model itself. We propose an alternative path to MSE functionality using the parameterized SS model itself as the OM. The goal of this project is to build the capacity to easily turn any SS model into an operating model with as much of the OM engine as possible operating within SS, thereby making OM output an innate capacity of SS. Additional parts of the MSE control logic will be developed as an R package called SSMSE. This approach would allow the complexity of the MSE to grow with SS, leveraging its future development, and innately linking the MSE and stock assessment processes. Our purpose in this presentation is to describe the project, elicit input on desirable outputs and properties of OMs and to solicit a review panel of likely users and previous/existing MSE/SS developers to suggest important features, beta test software developments, and review the final products.

Jeremy McKenzie: The role of simulation modelling in fisheries research: future needs and requirements
https://www.youtube.com/watch?v=ht2f2APdfZ0

Simulation modelling is important to fisheries management and research in two key areas: management strategy evaluation (MSE); estimation-model performance evaluation. Although most of the common stock assessment modelling platforms can be configured for data simulation, greater insight into estimation model performance is gained when the simulated assessment data has been generated by a computationally different, and more "real world" complex, operating modelling platform. In this talk I will discuss some of the key roles and feature requirements of operating model simulators in fisheries research. I will compare the pros and cons of two currently in-use simulation modelling platforms: a partition modelling platform (SPM) and an agent-based modelling platform and pose the question: how much simulation capacity should we build into the next generation of fisheries models?

Alfonso Perez-Rodriguez: Multispecies modelling: estimation of reference points and assessment of joint HCRs that take into consideration ecological interactions.
https://www.youtube.com/watch?v=hqr3bAvGBBo

The multispecies tier is an essential part of the Northwest Atlantic Fisheries Organization (NAFO) roadmap for an Ecosystem Approach to Fisheries management. The European Union launched in 2017 the project SC05 "Multispecies Fisheries Assessment for NAFO" with the intention of identifying the potential alternatives to implement a multispecies approach in NAFO, with the Flemish Cap as a case study. In this paper, an MSE framework is developed, with GadCap (cod, redfish and shrimp Gadget multispecies model in the Flemish Cap) as operating model within the a4a FLR MSE framework. Reference points and Harvest Control Rules (HCR) were designed considering the multispecies interactions. Finally, traditional single species and new multispecies HCRs were assessed from the precautionary and MSY perspectives. The results suggest that HCRs designed under a single species approach are not precautionary for all the stocks and that it is not possible maintaining the 3 stocks above B_{lim} at the same time due to strong trophic interactions. Disregarding one stock may allow finding precautionary multispecies reference points for the other stocks. Precautionary HCRs for two stocks at once were only found when shrimp SSB in relation to B_{lim} was disregarded. The results showed that the two stages HCRs for cod reduces predation and increases probability of cod and redfish being above B_{lim}. This result supports that alternative two stage HCRs, or some other HCRs with designs different to the traditional single hockey stick, may increase the possible combinations of fishing strategies in system with multiple and interacting exploited stocks.

Piera Carpi, Allan C. Hicks, Ian J. Stewart, David T. Wilson: Insights on dynamic reference points and their importance for next generation stock assessment models.
https://www.youtube.com/watch?v=oDViadvuuhY

Non-anthropogenic influences on a fish population are important drivers of both dynamics and productivity and are often a major cause for changes in recruitment strength, growth, and natural mortality. However, there are only few examples where environmental components are formally included in fisheries management: for many stocks, the current management relies on reference points based on a theoretical unfished equilibrium stock size, which may lead to an inconsistency between the current potential of the stock and the management objectives. A dynamic reference points approach, on the other hand, will help to disentangle the effect of fishing from the effect of environmental forcing, given that the underlying mechanisms for the main changes in growth or productivity have been identified.

In recent years, the International Pacific Halibut Commission (IPHC) has revised its harvest strategy policy to use a dynamic measure of stock status rather than assuming a static unfished equilibrium biomass defined for a specific recruitment regime and growth pattern.

We first discuss the differences between candidate dynamic reference points: 1) “dynamic time-series reference points”, calculated for each year of the stock assessment using the past observations of recruitment deviations (and comprising a special case of the ‘moving window’ approach), and 2) “dynamic equilibrium reference points”, calculated using constant equilibrium recruitment and constant productivity. Then, the Pacific halibut stock is used as an example to i) investigate the variability of reference points, such as Maximum Sustainable Yield (MSY), over time, ii) understand the influence of different types of uncertainty on dynamic reference points, and iii) discuss the importance of incorporating dynamic reference points as part of a general stock assessment framework. Three different approaches were implemented. First, an equilibrium model was used to investigate changes in dynamic equilibrium reference points given different assumptions about virgin recruitment, steepness, weight at age, selectivity, and natural mortality. The stock synthesis framework was then used to investigate estimates of past and current dynamic equilibrium reference points. Finally, a simulation framework was implemented to investigate the variability in dynamic equilibrium reference points given variability in parameters, productivity, fisheries, and structural assumptions.

Given the potential benefits of dynamic reference points, we suggest that a clear characterization of candidate methodologies is important to ensure analysts are comparing the same quantities. Further investigation into how best to incorporate dynamic reference points calculations into stock assessment models and which are most appropriate will rely on next-generation models capable of calculating a range of alternatives.

Nathan Vaughan: A decision support tool for incorporating management impacts into stock assessment projections
https://www.youtube.com/watch?v=YM72bJR-djs

We present a stock assessment decision support tool (DST) that extends the existing projection capabilities of the stock synthesis integrated assessment model (SS). The DST provides a simple graphical user interface that automates the incorporation of recent landings data and future management changes into the calculation of equilibrium sustainability benchmarks and annual overfishing limits. Specifically, DST users can: input recent catch history; adjust allocations between fleets; assign equilibrium benchmark targets; and change future fleet selectivity parameters. The ability to incorporate proposed management actions in the estimation of overfishing limits (OFLs) is a first step towards full dynamic feedback between the assessment and management processes. Coupling the implementation of management with the estimation of OFLs will: improve the robustness of OFL estimates; provide best scientific advice regarding the expected impacts of management actions; and improve managers and stakeholders understanding of the complex tradeoffs inherent to all management options. Allowing non-experts to easily explore the impacts of different management actions could increase public trust and engagement in the stock assessment process. The DST has been utilized within the U.S. to inform management decisions for Gulf of Mexico fisheries and internationally by assessment scientists at ICCAT to produce Bigeye and Yellowfin tuna projections. Future development goals include: additional data inputs such as indices of abundance to enable simplified interim assessment advice to be produced from partial data; and utilizing the DSTs projection interface capabilities to develop an SS based management strategy evaluation package. The DST project represents an ongoing attempt to integrate the stock assessment and management/regulation processes; achieving this goal should be a desired feature of future advanced stock assessment methods.

Ernesto Jardim, Manuela Azevedo, Jon Brodziak, Liz Brooks, Kelli Faye Johnson, Nikolai Klibansky, Coilin Minto, Colin Millar, Iago Mosqueira, Richard Nash, Paris Vasilakopoulos, Brian Wells: Operationalizing model ensembles to provide scientific advice for fisheries management.
https://www.youtube.com/watch?v=TxZkwIifeRo

Providing scientific advice to fisheries managers can be a risky activity! It's not uncommon that a model which was working perfectly fails to properly fit an additional year of data, or to find that projections made in the past did not materialise when new information was made available. Furthermore, when fitting a model it is common to deal with conflicting pieces of information or data revisions. All of these have the insidious tendency to ruin a seemingly good assessment model. Scientists deal with very complex systems, with many unknown or poorly understood processes and limited information, which make advisory tools sensitive to alternative system representations, model assumptions or new data. Our approach to mitigate the potential lack of robustness and instability of fisheries advice is to expand its basis to integrate structural uncertainty using model ensembles. There are two main reasons to use model ensembles (i) to include structural uncertainty captured by differences across models of the same system; and (ii) to integrate across initial conditions and process errors in projections or sensitivity runs. This paper discusses and speculates about the utility and implementation of model ensembles for scientific advice to fisheries management. We discuss ensemble utilization, ensemble types, weighting metrics, model space and model expansion. We make the case for using ensembles in three main situations: (i) to estimate stock status, (ii) to set future fishing opportunities, and (iii) to build operating models for management strategy evaluation.

A3.6 Diagnostics

Felipe Carvalho, Henning Winker, Laurence Kell, Dean Courtney, Massimiliano Cardinale, Dawit Ghebrehiwet, Michael Schirripa, Maia Kapur, Kevin R. Piner, and Mark N. Maunder: A cookbook for using model diagnostics in integrated stock assessments.
https://www.youtube.com/watch?v=u4KtHYmd08U

This study provides practical guidelines for implementing selected diagnostic tools that identify data conflicts and predictive skills in integrated age-structured stock assessment models. Conflicting signals are a major issue when attempting to integrate diverse data series within a complex model structure. Conflicts among data sets are often a symptom of model misspecification, evident as poor model fits, and can affect the estimates of important parameters and their derived quantities. Consequently, diagnostics that identify inadequate fits to data or conflicts among data components can be used as starting points to identify potential model misspecification and possibly inform alternative model formulations. Here we explore the utility of implementing a suite of diagnostic tools simultaneously in an integrated flow diagram prior to drawing any conclusions about model misspecification and before any data sets or models are removed from consideration. Particular emphasis is placed on evaluating predictive skills by incorporating retrospective analysis and hind cast cross-validation techniques as important diagnostic steps. By first implementing a complete set of diagnostics, one can get a much better sense of the interactions between them, more so than examining individual results in isolation. Example applications are provided from recent Regional Fishery Management Organization (RFMO) data-moderate length- and age-structured stock assessments using Stock synthesis. We use the Stock Synthesis modelling framework as a case study because of the availability of the r4ss package, a collection of R functions that automates visualizations of model summaries. The proposed suite of diagnostics comes with a generic set of R diagnostic tools and graphical visualizations.

A3.7 Parameter estimation

Anders Nielsen: The role of random effects in next-generation stock assessment models
https://www.youtube.com/watch?v=xbqvrhGLVfI

For most of the long history of fish stock assessment models, random effects have not explicitly been part of the model formulation. The obvious question is therefore if it is important to implement next-generation stock assessment models to support random effects at all?

State-space models are models where random effects are used to formulate stochastic processes which are unobservable but related to the observed quantities. From a theoretical point of view, the state-space framework is a perfect match to the typical assessment problem. However, in practice, the size and complexity of the applied assessment models, combined with the fast run times required for them to be operational, has in the past hindered the widespread use of state-space assessment models.

Recent advances in algorithms, software, and computing power has greatly reduced the practical concerns w.r.t. using state-space assessment models and they are increasingly being applied.

Here I will present some information about what random effects are used for in current assessment models (e.g.,~missing observations, time-varying quantities, and defining correlations) and outline where we can expect to see them further used in next-generation assessment models (e.g.,~prediction, linking stocks, and including spatial data).

I will argue that random effects simplify and generalizes many aspects of formulating assessment models and opens up promising paths to further advance assessment models, and hence that good support for random effects should be high on the feature list for next-generation assessment models.

Brian C. Stock and Timothy J. Miller: The Woods Hole Assessment Model (WHAM): a generalized state-space age-structured stock assessment model that can include environmental effects on population processes
https://www.youtube.com/watch?v=M1x4Nv4Ibio

The Northeast Fisheries Science Center (NEFSC) is developing the Woods Hole Assessment Model (WHAM), which builds off ASAP, its ADMB-based statistical catch-at-age (SCAA) model. WHAM has many similarities to ASAP, including the input data file structure, and many of the plotting functions for input data, results, and diagnostics are modified from ASAP code. WHAM is written in TMB and can be configured to estimate a range of assessment models, from a traditional SCAA model with recruitments as (possibly penalized) fixed effects, SCAA with recruitments as random effects, or a full state-space model with abundance at all ages treated as random effects. We have distributed WHAM as an R package on GitHub with the intent to follow guidelines for free and open-source software of the NOAA Fisheries Toolbox and the Linux Foundation Core Infrastructure Initiative. R functions allow users to load input data files, specify model options, fit the model in TMB, conduct retrospective and one-step-ahead residual analysis, check convergence, plot diagnostics and results, and compare alternative models.

The primary factors motivating WHAM development are the state-space formulation and incorporation of environmental covariates on time-varying population processes. Several recent applications to Northwest Atlantic stocks have shown that these models can outperform traditional SCAA models in terms of reduced bias, retrospective pattern, AIC, and uncertainty in reference points. In addition, large changes in oceanographic conditions are occurring on the Northeast U.S. shelf, and WHAM has been designed to evaluate how these changes may affect productivity of the commercially important stocks in the region. While standard assessment models can account for environmentally driven shifts in productivity by estimating time-varying population parameters, WHAM explicitly models the environmental drivers, which can reduce uncertainty and refine projections of stock status. We describe case studies highlighting WHAM’s capabilities and share plans for future development.

Noel Cadigan: Approaches for modelling landings and catch age composition information in state-space stock assessment models
https://www.youtube.com/watch?v=of6IteJ6O18

I propose that landings information should be modelled using a different likelihood function than catch age composition information in an integrated state-space stock assessment model. This is important for some Northwest Atlantic stocks with international fisheries where uncertainty about landings levels may be high in some time-periods. However, we have little knowledge about the accuracy of landings, and in this case I suggest that a catch bounds approach using a censored likelihood is a reasonable strategy for modelling this type of information. For catch age composition information, I investigate several approaches that depend on whether information on age composition sample sizes is available (i.e., Multinomial, Dirichlet-Multinomial, Log-Gaussian Cox Process) or not (Additive and Multiplicative Logistic Normal distribution). Heuristic results favour the Log-Gaussian Cox Process when sample sizes are approximately known, and the Multiplicative Logistic Normal otherwise.

Saang-Yoon Hyun, Jinwoo Gim, and Kyuhan Kim: Size-based and state-space production models for a fish stock assessment
https://www.youtube.com/watch?v=CFDyq3aU764

Although an age-structured model is most appreciated in fish stock assessments, it cannot be applied to a fish stock whose age information is not available. Alternative models range from a production model to a size-based model, depending on available data. When body size data such as lengths or weights from a temporal range are available with fishery catch and effort data, a size-based model would be an excellent option. On an extreme case when available data are limited only to fishery catch and effort data, a production model should be applicable. Given data about the Korean chub mackerel stock (*Scomber japonicus*) whose body lengths from 2000 to 2017 and fishery catches and efforts from 1996 to 2017 were available, we implemented size-based and production models in ADMB and ADMB-RE (Random Effect module). We extended the evolution of size-based models from Cohen & Fisherman (1980), Deriso and Parma (1988) to Quinn et al. (1998) for the former while we applied a Bayesian state-space production for the latter. We will present detailed methods of those two approaches, and discuss results from them, also seeking workshop participants' ideas about how to systematically synthesize those two approaches.

Craig Marsh: My Biased experience of using the generalised packages CASAL & Casal2 vs standalone ADMB and STAN models for stock assessments.
https://www.youtube.com/watch?v=OaqTDuyfOPA

My background has been as an applied statistician pretending to be a coder and convincing myself I knew how to help modify and enhance the generalized package Casal2; so I am fairly proficient at setting up and debugging stock assessments using Casal2. Which actually made me the worst tester for Casal2!

I want to talk about my experience with interfaces when setting up and running assessments, comparing the generalised package Casal2 with standalone programs like ADMB/STAN.

The topics I want to talk about regard, input configurations, error messages, tweaking models from working group recommendations, user community and more. I will spend a little bit of time going over the Casal2 user interface to demonstrate its capabilities, which will hopefully lead into a discuss whether people think that is a useful interface, too much, not enough etc.

A3.8 User interface and good practices defaults

Arni Magnusson and Colin Millar: ICES Transparent Assessment Framework (TAF)
https://www.youtube.com/watch?v=-xGi2kGNP_Y

In 2016, ICES began the development of a new framework that organizes all stock assessments in a standard format that is open and fully reproducible. ICES releases advice for around 250 stocks per year, based on a very wide variety of models and related methods. The ICES Transparent Assessment Framework (TAF) was launched in December 2018, it is now operational, and the core functionality is fully implemented.

The TAF workflow is very general, asking stock assessors to write four main scripts that run the assessment. The scripts are run sequentially to run the entire assessment from data preparation all the way to result tables and figures: data.R, model.R, output.R, and report.R. Each script can be split into separate steps: data_construct.R, data_export.R, model_fit.R, model_forecast.R, etc. All required software and underlying data files are archived to support long-term reproducibility. Once the advice is released, anyone can browse the analysis online or download and run on their own computer.

A key objective in TAF is to allow the stock assessor to use any model and methods. The only requirement is that the model can be run from an R script. Models are typically run in R using a dedicated R package or by launching an external application using the system() command or similar approaches.

Stock assessors around the world are currently making their assessments more open and reproducible. In this talk we summarize a few key messages to software developers and stock assessors, based on TAF experiences.

Bjarki Þór Elvarsson, Jamie Lentin and Pamela J. Woods: The RGadget environment: A tidyverse inspired approach to model development work flow
https://www.youtube.com/watch?v=RDej46_5IlE

Various frameworks and software packages have been developed in recent years to aid in the process of assessing marine resources. One in particular, the Gadget framework, allows for the development of statistically testable models of marine multi-species ecosystems. Gadget allows for the definition of age- and length-structured models that can take multi-fleet and mixed fisheries into account. Gadget has proven to be useful in cases where data on the resource is scarce or has notable gaps. Gadget's flexibility does however allow the user to define a quite varied set of models and output, making it cumbersome to manage and compare different model versions.

This has led to the development of two related R-packages, Rgadget and MFDB, that compose the Rgadget environment. RGadget is an R-package built to aid in the development and testing of models built using the Gadget framework, whereas the MFDB package is a database tool that stores data needed for ecosystem models in a minimally aggregated format that can be extracted in the form required for a Gadget model. The aim of these packages is to facilitate rapid model development, and as such, it includes several tools useful for the model developer. These include functions to build models and flexibly extract data, a heuristic to assign weights to disparate datasets used by a typical Gadget model and routines that organize output and provide model diagnostic figures. Model variants can be created relatively easily by simple update commands, thus allowing for better project management and traceability.

This presentation will briefly introduce Gadget and how one can set up a Gadget model using the Rgadget environment.

Ian Taylor: Processing and exploring assessment model output: lessons learned from a decade of work on the r4ss package
https://www.youtube.com/watch?v=PB70VtZN0QU

The r4ss package is a widely used collection of R functions that has been developed in parallel with Stock Synthesis for more than ten years. Thanks to a long history of improvements, contributions from 27 people, and feedback from the large user base, it has become a powerful tool to efficiently explore assessment model results, illustrate assessment reports, and manipulate model inputs. The choice of R, which is so widely used by fisheries scientists, has been instrumental to its success. However, behind the scenes the package is a bit of a mess. The accumulated technical debt of years' worth of edits to aging code have made adapting to evolving uses of the package more difficult. I will discuss lessons learned from a decade of work on this project for processing output from the next generation of assessment models.

A3.9 Coordination, project planning, hosting, and funding

Jennifer Ferreira: Agile Software Development – What is it and do we want it?
https://www.youtube.com/watch?v=OJqBkPiD4hM

Agile software development is an umbrella term for a set of frameworks and practices that are built on the values and principles expressed in the Manifesto for Agile Software Development. In the twelve years that I have been working with Agile teams, I have seen successes and failures in organisations big and small, here in New Zealand and abroad. In this talk I will highlight some of those experiences alongside industry best-practices. I will address some of the challenges and opportunities facing teams who are new to Agile or who have yet to embark on their Agile journey.

Craig Anslow: Satisfaction, Practices, and Influences in Agile Software Development
https://www.youtube.com/watch?v=o6WAt6C3ov0

The principles behind the Agile Manifesto begin with "Our highest priority is to satisfy the customer…". It also states that Agile projects should be built around motivated and self-organized teams, which might also lead to more satisfied developers. Several studies indeed report an increased job satisfaction by anecdotal evidence.

In this talk we address the topic of satisfaction by in-depth analysis of the results of a nationwide survey about software development in Switzerland. We wanted to find out if satisfaction depends on the applied development method, and, more concrete, how satisfaction relates to other elements in the development process, including the use of various practices, and the influences on business, team and software issues.

We found that higher satisfaction is reported more by those using Agile development than with plan-driven processes. We explored the different perspectives of developers and those with a management role and found a high consistency of satisfaction between Agile developers and Agile management, and big differences with using working plan-driven methods. We found that certain practices and influences have high correlations to satisfaction, and that collaborative processes are closely related to satisfaction, especially when combined with technical practices. Applying recursive partitioning, we found which elements were most important for satisfaction, and gained insight about how practices and influences work in combination.

We also explored the relationship between satisfaction and personal experience with Agile development. Our results in this analysis are principally descriptive, but we think they can be a relevant contribution to understand the challenges for everyone involved in Agile development and can help in the transformation to Agile.

Hilary Oliver: The Cylc Workflow Engine: Sustaining a Collaborative Scientific Software Project
(no video available)

Cylc is a workflow orchestration engine for complex cycling systems. Since its release by NIWA on an Open Source license 8 years ago it has been widely adopted to drive environmental forecasting applications around the world, and the project is still evolving rapidly. Cylc has precious little to do with fish, so Hilary Oliver (Cylc project lead) will try to focus on general aspects of collaborative scientific software development, and what it takes to successfully grow and sustain a project like this over time.

Johnoel Ancheta: ADMB Project Infrastructure and the Next Steps
https://www.youtube.com/watch?v=W264iS8vzsQ

Automatic Differentiation Model Builder (ADMB) software is used for developing and fitting general nonlinear statistical models to analyse data. The software was developed by Dr. Dave Fournier who formed Otter Research Ltd to commercially develop ADMB. Eventually, ADMB became open-source and was made freely available to the public. The ADMB Project was created to continue the development and maintenance of ADMB as an open-source project. The user and developers for ADMB are worldwide. An open-source infrastructure was needed to support the community for collaboration and information gathering.

The presentation will describe the ADMB Project infrastructure, tools, and services, and explain how and why they are used for the project. Results from the ADMB user survey administered at the CAPAM workshop will also be presented. As the software and hardware systems continue to change, the next steps will describe the future goals and tasks for the ADMB Project.

Yoann Ladroit: Journey of an open source software for fisheries acoustics
https://www.youtube.com/watch?v=D-AUSyHb4OQ

ESP3 is an open-source software package for visualizing and processing fisheries acoustics data with attention to reproducibility, consistency, and efficiency. It was started at sea during a trip to Antarctica in 2015 and its first version was released under MIT license in 2017. It has reached a significant number of people since then and has been popular with students and researchers due to the availability of the code. It is scientific software and its outputs are used to produce absolute or relative biomass indices for stock assessment models or spatio-temporal ecosystem studies, making it critical to have rigorous testing procedures in place.

The project itself was started “accidentally” due to the difficulty of reproducing results obtained by our previous generation of software using commercially available tools. From this point, it grew organically at first with minimal planning, until it became evident that a strategy had to be put in place to make it a viable long-term project and in order to better manage its growing source code. We moved forward by relying heavily on organized direct feedback from a small number of key users and by using modern code versioning and sharing tools. The interaction with users has proven key to help us introduce, test and document new features successfully, and has led to very fruitful ongoing collaborations.

Kelli Johnson: Lessons learned from the ss3sim project on sharing, hosting, and maintaining simulation code
https://www.youtube.com/watch?v=cE4AmiE-1oA

Code to simulation test stock assessment models is plentiful but is often generated for a specific project or stock assessment, leading to a lack of sharing and maintenance. Thus, ss3sim was created to facilitate reproducible, flexible, and rapid end-to-end simulation testing of stock assessment models that utilize the Stock Synthesis framework. As the codebase for ss3sim grew so did demands for maintenance and development.

Implementing automated testing procedures using the testthat package was helpful, but the package benefited the most from user feedback. Through the use of github, users provided valuable insight regarding the helpfulness, or lack thereof, of warning messages; functions that weren’t quite doing what they were intended to do or better ways to get the job done; and suggestions to ease the learning curve for new users.

We highlight some of the choices that were made regarding its initial simplicity that still seem like good ideas, mistakes that could have been avoided had we known better, and future avenues of development.

APPENDIX 4: MEETING AGENDA

- ### Monday

Introduction and welcome

09:00 Welcome (Pamela Mace) and introductions

Coding philosophies, software structure, and underlying language base

09:30 Matthew Supernaw – Engineering Practices for Maintainable Software (60)
10:30 Morning tea
11:00 Corinne Bassin – Enabling Successful Onboarding of Scientific Tools Via Development Best Practices (30)
11:30 Scott Rasmussen – Coding CASAL2 (30)
12:00 Lunch
13:00 Ernesto Jardim – A modular framework for the generic application of fisheries management strategy evaluation. (30)
13:30 Wen-His Yang – Doing Stock Assessment Using ADMB, Stock Synthesis or TMB? A Case Study on the Queensland Saucer Scallop. (30)
14:00 Discussion: Coding philosophies, software structure, and underlying language base (60)
15:00 Afternoon tea

Stock assessment model features

15:30 Momoko Ichinokawa – Stock assessment model in Japan: past to present (15)
15:45 Akira Hayashi – Stock assessment model in Japan: future perspective (15)
16:00 Kota Sawada – Case studies of the local stock assessment in the Northwest Pacific: difficulties in the stock assessment for seamount bottom fisheries (15)
16:15 Shin Fukui – Case studies of the local stock assessment in the Northwest Pacific: application of robust regression in estimating stock-recruitment relationship (15)
16:30 Jemery Day – Stock assessment issues in South Eastern Australia (30)

- ### Tuesday

09:00 Andre Punt – Essential Features of the Next-Gen Integrated Assessment (60)
10:00 Erik Williams – Beaufort Assessment Model (BAM): Lessons Learned From Twenty Years of Software Development (30)
10:30 Morning tea
11:00 Allan Hicks – Needs from a next generation general modelling framework to support the future of stock assessment and MSE at IPHC (30)
11:30 Divya Varkey (presented by Jonathan Babyn) – Hybrid: a modelling framework to sidestep structural uncertainty in models (30)
12:00 Chantel Wetzel – Moving up the assessment ladder: A flexible and integrated approach to modelling data-limited stock assessments (30)
12:30 Lunch

Groups/partitions

13:30 Nick Davies – Partitioning in MULTIFAN-CL in respect of space, tagged populations, species, stocks, and gender - coding implementation and recommendations (60)
14:30 Richard McGarvey – Will age-and-length based modelling permit broader application of the next-generation fishery assessment model? (30)
15:00 Afternoon tea

15:30 Jim Ianelli – Assessment developments including climate enhanced multi-species models from the North Pacific. (30)
16:00 Steve Saul – Are Agent-Based Approaches the Future of Fishery Management? – applying the Poseidon model to the Eastern Pacific Tropical Tuna Fishery (30)
16:30 Discussion: Stock assessment model features 1 (60)

- **Wednesday**

Observation models

09:00 John Feenstra – A framework for multi-year Leslie-Davis depletion modelling and its use as a stock assessment model feature. (30)
09:30 Rich Hillary – Integrating conventional and electronic tagging data into the next generation of stock assessment models (30)
10:00 Hans Skaug – Software for Close-kin Mark-recapture (30)
10:30 Morning tea
11:00 Robin Thomson – Application of Close-Kin Mark-Recapture (30)
11:30 Mark Chambers – Implications of entrainment for fisheries stock assessment (30)
12:00 Lunch
13:00 Jonathan Babyn – A New Approach to Generating Spatial Age-Length Keys based on Using a Gaussian Field Approximation with Support for Physical Barriers (30)

Management quantities

13:30 Rick Methot – Stock Synthesis Completes the Cycle: Assessment - Management Quantities - Projections - MSE / Propagating Variance in Assessment-Linked Projections (60)
14:30 Skyler Sagarese – Management strategy evaluation made operational with Stock Synthesis (30)
15:00 Afternoon tea
15:30 Jeremy McKenzie – The role of simulation modelling in fisheries research: future needs and requirements (30)
16:00 Piera Carpi – Insights on dynamic reference points and their importance for next generation stock assessment models. (30)
16:30 Nathan Vaughan – A decision support tool for incorporating management impacts into stock assessment projections (30)

- **Thursday**

09:00 Alfonso Perez-Rodriguez – Multispecies modelling: estimation of reference points and assessment of joint HCRs that take into consideration ecological interactions
09:30 Ernesto Jardim – Operationalizing model ensembles to provide scientific advice for fisheries management. (30)

Diagnostics

10:00 Felipe Carvalho (presented by Dean Courtney) – A cookbook for using model diagnostics in integrated stock assessments. (30)
10:30 Morning tea
11:00 Discussion: Stock assessment model features 2 (60)
12:00 Lunch

Parameter estimation

13:00 Anders Nielsen – The role of random effects in next-generation stock assessment models (30)

13:30 Brian Stock – The Woods Hole Assessment Model (WHAM): a generalized state-space age-structured stock assessment model that can include environmental effects on population processes (30)
14:00 Noel Cadigan – Approaches for modelling landings and catch age composition information in state-space stock assessment models (30)
14:30 Saang-Yoon Hyun – Size-based and state-space production models for a fish stock assessment (30)
15:00 Afternoon tea
15:30 Craig Marsh – My Biased experience of using the generalised packages CASAL & Casal2 vs standalone ADMB and STAN models for stock assessments. (30)
16:00 Discussion: Stock assessment model features 3 (60)

- **Friday**

User interface and good practice defaults

08:30 Arni Magnusson – The ICES Transparent Assessment Framework
09:00 Bjarki Elvarsson – The RGadget environment: A tidyverse inspired approach to model development workflow (30)
09:30 Ian Taylor – Processing and exploring assessment model output: lessons learned from a decade of work on the r4ss package (30)
10:00 Discussion: User interface and good practices defaults (30)
10:30 Morning tea

Coordination, project planning, hosting, and funding.

11:00 Jennifer Ferreira – Agile Software Development – What is it and do we want it? (30)
11:30 Craig Anslow – Satisfaction, Practices, and Influences in Agile Software Development (30)
12:00 Lunch
13:00 Hilary Oliver – The Cylc Workflow Engine: Sustaining a Collaborative Scientific Software Project (30)
13:30 Johnoel Ancheta – Hosting ADMB (30)
14:00 Yoann Ladroit – Journey of an open source software for fisheries acoustics (30)
14:30 Kelli Johnson – Lessons learned from the ss3sim project on sharing, hosting, and maintaining simulation code (30)
15:00 Afternoon tea
15:30 Discussion: Coordination, project planning, hosting, and funding (60)

Workshop summary

16:30 General discussion (30)

APPENDIX 5: WORKSHOP PARTICIPANTS

First name	Last name	Affiliation	State	Country	Email
Jemery	Day	CSIRO	TAS	Australia	jemery.day@csiro.au
Ainslie	Denham	DPIRD, Western Australia	WA	Australia	ainslie.denham@dpird.wa.gov.au
Cathy	Dichmont	CDC Consulting, Australia	QLD	Australia	cathydichmont@gmail.com
John	Feenstra	SARDI, South Australia	SA	Australia	john.feenstra@sa.gov.au
Malcolm	Haddon	CSIRO	TAS	Australia	malcolm.haddon@gmail.com
Alex	Hesp	DPIRD, Western Australia	WA	Australia	Alex.Hesp@dpird.wa.gov.au
Rich	Hillary	CSIRO	TAS	Australia	Rich.Hillary@csiro.au
Robyn	Lovett	Fisheries Queensland	QLD	Australia	Robyn.Lovett@daf.qld.gov.au
Rick	McGarvey	SARDI, South Australia	SA	Australia	richard.mcgarvey@sa.gov.au
Bruce	Taylor	Modelling and Data Solutions	VIC	Australia	modellingsolutions@fastmail.fm
Robin	Thomson	CSIRO	TAS	Australia	robin.thomson@csiro.au
Wen-Hsi	Yang	University of Queensland	QLD	Australia	w.yang@uq.edu.au
Santiago	Quijano	State University of Sao Paulo		Brazil	santiago.m.quijano@unesp.br
Jonathan	Babyn	Dalhousie University	NL	Canada	jn805248@dal.ca
Noel	Cadigan	Memorial University of Newfoundland	NL	Canada	Noel.Cadigan@mi.mun.ca
Divya	Varkey	Fisheries and Oceans Canada	NL	Canada	divya.varkey@dfo-mpo.gc.ca
Anders	Nielsen	Technical University of Denmark		Denmark	an@aqua.dtu.dk
Arni	Magnusson	ICES		Denmark	Arni.magnusson@ices.dk
Colin	Millar	ICES		Denmark	colin.millar@ices.dk
Bjarki	Elvarsson	Marine and Freshwater Research Institute		Iceland	bjarki.elvarsson@hafogvatn.is
Paula	Silvar	Galway-Mayo Institute of Technology		Ireland	paula.silvarviladomiu@research.gmit.ie
Shin	Fukui	National Res. Inst. of Far Seas Fisheries		Japan	shinfukui@affrc.go.jp
Akira	Hayashi	National Res. Inst. of Far Seas Fisheries		Japan	akirahayashi@affrc.go.jp
Momoko	Ichinokawa	National Res. Inst. of Far Seas Fisheries		Japan	ichimomo@fra.affrc.go.jp
Kota	Sawada	National Res. Inst. of Far Seas Fisheries		Japan	kotasawada@affrc.go.jp
Nicholas	Ducharme-Barth	Pacific Community (SPC)		New Caledonia	nicholasd@spc.int
John	Hampton	Pacific Community (SPC)		New Caledonia	johnh@spc.int
Matthew	Vincent	Pacific Community (SPC)		New Caledonia	matthewv@spc.int
Alfonso	Perez-Rodriguez	Institute of Marine Research		Norway	alfonso.perez-rodriguez@hi.no
Hans	Skaug	University of Bergen		Norway	Hans.Skaug@uib.no
Teresa	A'mar	NIWA		NZ	teresa.amar@niwa.co.nz
John	Annala	Fisheries New Zealand		NZ	john.annala@mpi.govt.nz
Craig	Anslow	Victoria University of Wellington		NZ	craig@ecs.vuw.ac.nz
Tiffany	Bock	Fisheries New Zealand		NZ	tiffany.bock@mpi.govt.nz
Mark	Chambers	Pisces Research		NZ	mark.stanley.chambers@gmail.com
Matteo	Chiarani	University of Auckland		NZ	matteo.chiarini@niwa.co.nz

First name	Last name	Affiliation	State	Country	Email
Samik	Datta	NIWA		NZ	samik.datta@niwa.co.nz
Nick	Davies	Pacific Community (SPC)		NZ	nd200806@gmail.com
Jennifer	Devine	NIWA		NZ	jennifer.devine@niwa.co.nz
Matt	Dunn	NIWA		NZ	matt.dunn@niwa.co.nz
Alistair	Dunn	Ocean Environmental Ltd.		NZ	alistair.dunn@OceanEnvironmental.co.nz
Adele	Dutilloy	NIWA		NZ	adele.dutilloy@niwa.co.nz
Charlie	Edwards	CEESCAPE consulting		NZ	cescapecs@gmail.com
Jennifer	Ferreira	Victoria University of Wellington		NZ	jennifer@ecs.vuw.ac.nz
William	Gibson	Fisheries New Zealand		NZ	william.gibson@mpi.govt.nz
Steven	Holmes	NIWA		NZ	steven.holmes@niwa.co.nz
Simon	Hoyle	NIWA		NZ	simon.hoyle@niwa.co.nz
Rosie	Hurst	NIWA		NZ	rosemary.hurst@niwa.co.nz
Marco	Kienzle	NIWA		NZ	Marco.Kienzle@niwa.co.nz
Kyuhan	Kim	Victoria University of Wellington		NZ	kh2064@gmail.com
Yoann	Ladroit	NIWA		NZ	yoann.ladroit@niwa.co.nz
Adam	Langley	Trophia		NZ	adam_langley@xtra.co.nz
Kath	Large	NIWA		NZ	kath.large@niwa.co.nz
Pamela	Mace	Fisheries New Zealand		NZ	Pamela.Mace@mpi.govt.nz
Craig	Marsh	University of Auckland		NZ	craig.marsh10@gmail.com
Vidette	McGregor	NIWA		NZ	vidette.mcgregor@niwa.co.nz
Jeremy	McKenzie	NIWA		NZ	jeremy.mckenzie@niwa.co.nz
Andy	McKenzie	NIWA		NZ	Andy.McKenzie@niwa.co.nz
Hilary	Oliver	NIWA		NZ	hilary.oliver@niwa.co.nz
Marine	Pomarede	Fisheries New Zealand		NZ	marine.pomarede@mpi.govt.nz
Scott	Rasmussen	Zaita		NZ	scott@zaita.com
Gretchen	Skea	Fisheries New Zealand		NZ	gretchen.skea@mpi.govt.nz
Laura	Tremblay-Boyer	Dragonfly		NZ	laura@dragonfly.co.nz
Nathan	Walker	Fisheries New Zealand		NZ	nathan.walker@mpi.govt.nz
Saang-Yoon	Hyun	Pukyong National University		South Korea	shyunuw@gmail.com
Ernesto	Jardim	Joint Research Centre, EU		Spain	ernesto.jardim@ec.europa.eu
Johnoel	Ancheta	University of Hawaii	HI	USA	johnoel@hawaii.edu
Corinne	Bassin	NOAA	WA	USA	corinne.bassin@noaa.gov
Piera	Carpi	IPHC	WA	USA	piera.carpi@iphc.int
Dean	Courtney	SEFSC, NOAA	FL	USA	dean.courtney@noaa.gov
Allan	Hicks	IPHC	WA	USA	allan.hicks@iphc.int
Jim	Ianelli	NOAA	WA	USA	jim.ianelli@noaa.gov
Kelli Faye	Johnson	NOAA	WA	USA	kelli.johnson@noaa.gov
Maia	Kapur	University of Washington	WA	USA	kapurm@uw.edu

First name	Last name	Affiliation	State	Country	Email
Brian	Langseth	NOAA	HI	USA	brian.langseth@noaa.gov
Emily	Liljestrand	Michigan State University	MI	USA	liljest3@msu.edu
Patrick	Lynch	NOAA	MD	USA	patrick.lynch@noaa.gov
Mark	Maunder	IATTC	CA	USA	mmaunder@iattc.org
Rick	Methot	NOAA	WA	USA	richard.methot@noaa.gov
Andre	Punt	University of Washington	WA	USA	aepunt@uw.edu
Steve	Saul	Arizona State University - Ocean Conservancy	AZ	USA	Steven.Saul@asu.edu
Brian	Stock	NOAA	HI	USA	brian.stock@noaa.gov
Matthew	Supernaw	NOAA	FL	USA	matthew.supernaw@noaa.gov
Ian	Taylor	NOAA	WA	USA	ian.taylor@noaa.gov
Nathan	Vaughan	NOAA	FL	USA	nathan.vaughan@noaa.gov
Chantel	Wetzel	NOAA	WA	USA	chantel.wetzel@noaa.gov
Erik H	Williams	NOAA	NC	USA	erik.williams@noaa.gov
		Registered remote attendees			
Steven	Berukoff	IPHC	WA	USA	steven.berukoff@iphc.int
Claudio	Castillo-Jordan	University of Washington	WA	USA	claudioj@uw.edu
Kathryn	Doering	NOAA	WA	USA	kathryn.doering@noaa.gov
Xi	He	NOAA	CA	USA	xi.he@noaa.gov
Huiihua	Lee	NOAA	CA	USA	huihua.lee@noaa.gov
Ian	Stewart	NOAA	WA	USA	Ian@iphc.int
Timothy	Miller	NOAA	MA	USA	timothy.j.miller@noaa.gov