Towards A Multi-Purpose Framework For Tax-Benefit Microsimulation: Lessons From EUROMOD

Herwig Immervoll[1] and Cathal O'Donoghue[2]
[1] Social Policy Division, OECD, 2 rue André Pascal, 75775 Paris Cedex 16, France;
   email: herwig.immervoll@oecd.org
[2] Rural Economy Research Centre, Teagasc, National University of Ireland, Galway;
   email: Cathal.ODonoghue@teagasc.ie

ABSTRACT: Tax-benefit models provide tools for policy analyses that should enable researchers to focus their attention on formulating policy scenarios and analysing their effects. From the users' and the developers' points of view, numerous characteristics and features are desirable to maximise the model's usefulness. A model framework that offers generalised components essential for tax-benefit modelling while at the same time providing a large degree of flexibility in defining the specific parameters can be re-used for a multitude of modelling purposes. This paper discusses issues arising in the construction of such a general framework and illustrates possible approaches by reference to the the framework developed for construction of the EUROMOD tax-benefit model. EUROMOD is an integrated tax-benefit microsimulation model covering 15 (pre 2004) countries that are members of the European Union (Immervoll *et al*, 1999) as well as 4 of the New Member States. Implementing this many tax-benefit systems in one single consistent framework requires a robust yet flexible structure. The framework needs to reflect the basic structural characteristics of tax-benefit systems while leaving enough room for a diversity of particular instruments and rules. This paper outlines the general model framework adopted. We argue that, apart from its direct usefulness for EUROMOD, the framework has far wider applicability as a general approach to static tax-benefit microsimulation modelling.

1. INTRODUCTION

Tax-benefit microsimulation models (MSMs) have been widely used in many countries for a number of years. Recently there has been interest in carrying out cross-country comparative exercises, examining the performance of policy instruments in different countries. Because of the limitations of national specific models in this regard (See Callan and Sutherland, 1997), an integrated multi-country model, EUROMOD, has been developed by a consortium of teams in 19 EU countries. This paper describes the generalised software framework used to construct EUROMOD.

1.1 Tax-benefit microsimulation models

Tax-benefit microsimulation models (MSMs) are computer programs that calculate tax liabilities and benefit entitlements for individuals, families or households in a nationally representative micro-data sample of the population and are used by both governments and academics to study existing social and fiscal policies as well as policy reforms. As micro models, they take as the basis of their analytical framework the micro-level, typically individuals, families and households.[1] As simulation models, they simulate the detail of tax-benefit policy legislation and thus are in a position to evaluate existing tax-benefit policies and aid in the design of new individual schemes or entire systems. They calculate applicable amounts of each element of the tax-benefit system in the legal order so that interactions between different elements of the system are fully taken into account. The resulting taxes, benefits and income measures for each individual, family or household are weighted to provide results at the population level. MSMs have been developed and are in use in many OECD countries (see Sutherland, 1995 for a survey).

By incorporating the interactions of different elements of the tax-benefit system and by taking full account of the diversity of characteristics in the population, this approach allows a very detailed analysis of the revenue, distributional and incentive effects of individual policy instruments and the system as a whole. In particular, they provide a powerful means of performing "what if" analyses by allowing the analyst to manipulate all relevant parameters of the system such as tax rates, thresholds, amounts, income concepts, in an intuitive and user-friendly environment (see Redmond, Sutherland and Wilson, 1998).

In what follows we highlight the most important advantages of tax-benefit microsimulation models, drawing upon experience and examples from the EUROMOD model building project. The principal uses of MSMs are:

- simulating and evaluating existing policy;
- examining the effects of alternative policies;
- indicating pressures on behaviour created by existing and alternative policies.

Being based on representative household micro-data, MSMs have the capability of looking at the incidence of policy effects across the income distribution or across different types of individuals (for example, for different ages or by gender). They can also be used to examine the effectiveness and efficiency of policy in achieving various objectives such as the reduction of poverty or satisfying a set of desired redistributive properties.[2] Capturing the detail of actual legal rules, they allow for complex interactions between different policy instruments to be identified. Perhaps the most important use of MSMs is to examine the effects of alternative (both actual and hypothetical) policy reforms. They can thus be used to compute revenue effects of such reforms or to ensure policy reforms meet certain

budgetary requirements (such being revenue 'neutral'). Since both social protection programs and taxation instruments are modelled, one can explore both changes to social policy programs and means to finance them. As with existing policies, reform scenarios can be analysed in terms of their consequences for different population or income groups, the numbers of winners and losers (e.g., Atkinson *et al*., 2002; Piachaud and Sutherland, 2000).

While most tax-benefit models are open to behavioural extensions or can be embedded in a dynamic model that is able to 'age' the underlying population, many of them are of a 'static' nature. That is, they concentrate on 'first round' effects of policy reforms and disregard any behavioural consequences (e.g., in terms of labour supply or other choices that have an impact on household income) and inter-generational effects of policies. Even though such a 'static' simulation cannot measure the direct impact on behaviour of reforms, it can be used to determine the *pressures* on behaviour ('incentive effects') such as marginal tax rates and replacement rates.[3]

Apart from their main use as tools for analysing the effects of fiscal and social policy measures, these models are frequently used to impute tax- and benefit variables that are not elicited in the survey questionnaire. Weinberg (1999), for example, finds that, of seventeen data sources surveyed, taxes and/or social insurance contributions are imputed in five. Also, income variables in household survey data are frequently recorded net of income taxes and other levies on income, notably social insurance contributions. Using an iterative procedure together with the tax-benefit rules contained in MSMs, these models can be used to 'recover' gross incomes from net amounts by computing taxes and contributions which, after being deducted, result in the net amounts recorded in the original data (Immervoll and O'Donoghue, 2001b).

1.2 Multi-country tax-benefit models
Recent years have seen an increasing demand for tools to perform international studies, particularly in Europe. This has been driven by stronger socio-economic links between countries, a more comparative focus in policy analysis and through a desire to verify theories in different national settings. While building country specific MSMs is a complex and very resource intensive task, designing a multi-country model brings up entirely new issues. Previous research using cross-country microsimulation provides a guide to some of the approaches, opportunities and pitfalls.

This research can essentially be divided into three types:
   (a) comparisons using a single country MSM;
   (b) comparisons using different national models; and
   (c) models embedded in a consistent and comparative design.

Type (a) models apply different national systems on the population of a single country. Examples include Atkinson *et al.*, (1988), who compared the impact of replacing the French tax-benefit system with the UK system and O'Donoghue and Sutherland (1999) who studied different European family tax instruments using UK data. Abstracting from differences in population structures they can examine the direct impact of different national systems. However because policy instruments are designed with a particular national policy, or social context in mind, care must be taken in interpreting results that ignore these differences.

Type (b) models incorporate the differences in national populations and income distributions by using different national datasets. For example, Callan *et al.* (1996) compared the Irish and UK tax-benefit systems. Comparing two very similar systems in this way is a relatively straightforward process. Yet, due to large conceptual differences between national models in terms of their structure, definitions, scope and output, extending the analysis to cover additional countries while maintaining comparability proved to be insurmountable (Callan and Sutherland, 1997).

Type (c) models, which have recently started to be developed, try to address these difficulties. As a step towards an EU-wide model, a prototype six country model, Eur6 (Bourguignon *et al*, 1997) was constructed and has avoided many of the pitfalls associated with using different national models. As an integrated methodology, designed from the outset for comparative purposes, such models allow for flexibility in specifying the optimal data and modelling definitions. Using this type of model it is possible not only to compare national model results but also to pool them across countries, e.g., allowing for the position of individuals from different countries to be located within the multi-country income distribution.

1.3 Aims of the paper
This paper describes a microsimulation modelling framework that draws on the experiences from the development of the Eur6 prototype. Tax-benefit models need to be flexible enough to allow the simulation of far-reaching and *ex ante* unknown policy alternatives, while keeping the specification of relatively minor policy changes reasonably simple. The framework was first designed with the aim to run the national modules that make up EUROMOD, on one single platform. The principles developed, however, are more widely generalisable.

Creating a generic modelling framework presents several challenges. Each national tax-benefit system has a different structural logic and accommodating this structural diversity while keeping the model logically correct, robust and transparent to users is a major task. This is complicated further by the aim to be able to transfer policy instruments *between* countries to see, for example, what effects benefit X of country A would have if implemented in country B. Also, operating on a cross-national basis, there is a

need to be able to evaluate the differential effects that a common policy instrument would have in the different countries. In essence, these requirements mean that each instrument needs to have a common interface so that it can be taken out of its original context and "plugged" into another system.

We will consider some of these design issues and illustrate possible approaches by reference to the EUROMOD tax-benefit model. The structure of the paper is as follows. Section 2 discusses the general objectives and desirable features of MSMs. Using a single platform for a multitude of different tax-benefit systems requires the essential 'building blocks' of tax-benefit systems to be identified independently of any given country. Section 3 outlines our generalised structure of tax-benefit systems on which the modelling framework is based. Section 4 describes the principal design philosophy behind the framework, detailing some key concepts of tax-benefit model building. The implementation of these key concepts in the framework is described in section 5, while the last section concludes.

2. DESIRABLE FEATURES OF MICRO-SIMULATION MODELS

Previous studies which have focused on the overall design of tax-benefit models have mainly focused on data- and broader design issues (Hoschka, 1986; Merz, 1991; Citro and Hanushek, 1991a, 1991b; Sutherland, 1995). Our paper aims to build on previous work by operationalising accepted concepts in terms of the detailed computational design of MSMs. In addition, it will also discuss entirely new aspects which only arise in building an integrated multi-country model.

To set the scene as to how a microsimulation framework should be created for tax-benefit modelling, it is useful to start by discussing the potential demands placed on the model. Broadly the objectives can be classified under the following headings:
- Flexibility;
- Ease of use;
- Robustness;
- Transparency and consistency of structure and concepts;
- Maintainability; and
- Cost effectiveness.

It is the role of tax-benefit models to assist in the analysis of existing and alternative policy scenarios. Depending on the purpose of the analysis, scenarios to be analysed will often need to satisfy a number of requirements such as revenue neutrality, improving work incentives, reducing poverty, etc. Because tax-benefit systems are highly non-linear with a large number of parameters, the list of possible constraints is literally endless (see, for example, Sutherland, 1991). It follows that the design of tax-benefit models should be *flexible*, enabling users to

specify a wide range of different policy scenarios and make it easy to switch between scenarios. Hancock (1997) argues in a first analysis of the computing requirements for EUROMOD that flexibility is probably the most important corner stone of the computing strategy.

Although MSMs may in part be constructed by computer programmers, typical users will include economists, statisticians and social policy analysts in both academia and government. *Ease of Use* should ensure that all relevant features of the model are accessible to a wide range of users rather than just programmers. As a result, as much of the internal workings of the model should be accessible for users. At the same time, the complexity of the model should be organised hierarchically[4]. In other words, it should be possible to use 'basic' features of the model without having to know all the details about more 'complex' model components. This ensures that the model is powerful while at the same time being useful for users with different backgrounds or different analyses in mind.

To enable users to make changes to tax-benefit algorithms in a relatively safe environment, one would ideally have a standardised set of pre-fabricated building blocks that can be adapted for specifying the algorithm of every possible tax-benefit instrument without any need for major reprogramming. Each element should be a derivative of a basic template and should have the same type of input and output data structures. Only the core algorithm which determines the behaviour of the element would need to be element specific. Once users have become familiar with this structure, they can then adapt any tax-benefit algorithm without having to 'dig' through program code. However, in general there exists a trade-off between flexibility and *robustness*. It is technically possible to develop highly flexible elements that, through parameterisation, can be used for many different purposes. For example, a generalised tax allowance that can be used to construct all types of tax-allowance used in different countries. However, a very large number of parameters which attempt to provide for any potential use of an instrument may result in a model that is both difficult to use and is more prone to produce errors through mis-specification of parameters. In the case of our example, it may result in a tax allowance module that is very long and difficult to follow.

Once accustomed to the operation of one country module, users of a multi-country model need to be able to access the parameters of other countries' tax-benefit systems in a similar way. The multitude of necessary definitions and concepts (e.g., income taxes, fiscal units, sharing rules: see below) mean that *consistent* specification of relevant concepts across countries is essential. As highlighted in the previous section, simply lining up national models next to each other is not suitable as the design of national models will tend to reflect national priorities. Such
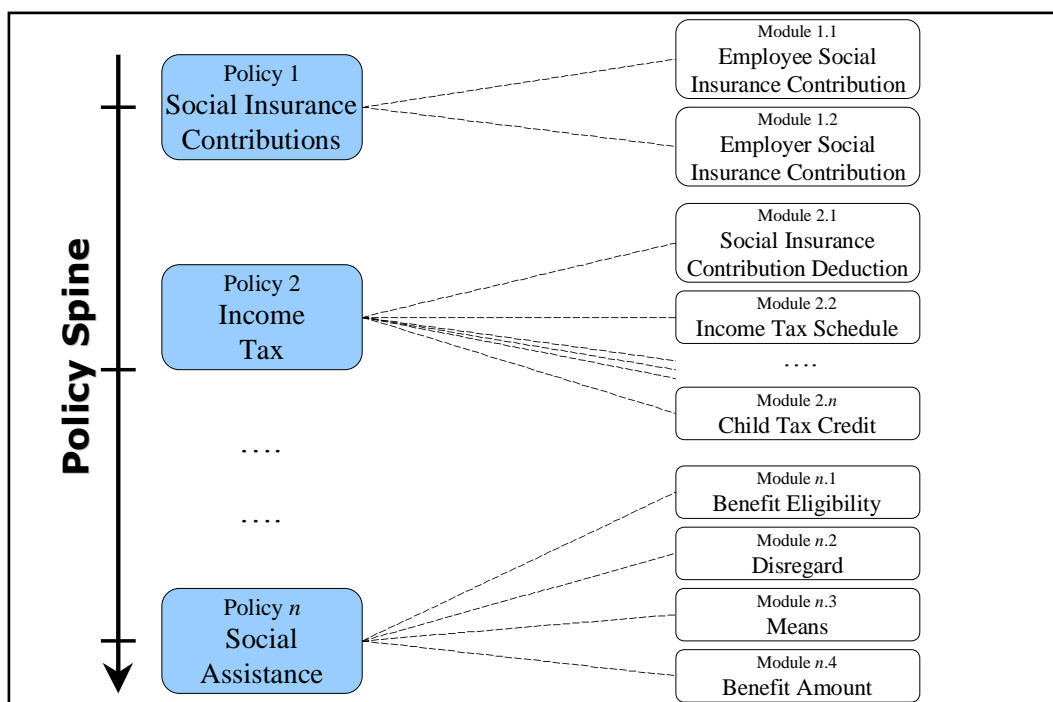
Figure 1 Structure of Tax-Benefit System

conceptual differences will make it hard or impossible to compare the results of different models and prevent a consistent specification of policy reforms. A generalised modelling framework should therefore allow *consistency* in the specification of different systems. Given the fact that a typical tax-benefit system of one single country encompasses thousands of parameters, all of these parameters need to be readily accessible and organised in a *transparent* manner.

In order to be able to contribute to the debate of contemporary policy issues, tax-benefit models will need *maintenance* on a regular basis. In addition to frequent revisions of tax-benefit rules, underlying micro-data will need to be updated regularly so that model results continue to be based on representative data. The model framework should therefore be able to access and organise different data sets with ease. In addition, the complexity of tax-benefit systems make validation of the logical correctness of the model an essential component of any model building project. Finding and analysing discrepancies between model results and reference figures should be aided by a model structure which allows tax-benefit algorithms to be 'traced' in order to find any modelling errors. Similarly, it is important to break complex algorithms down into manageable pieces that can be analysed separately.

The development of microsimulation models is a resource intensive process, involving the construction of a software environment to handle the data, policy simulation and output routines, the transformation and matching of existing micro-datasets and translation of tax-benefit laws into a computational framework (McCrae, 1999). Another important expense is updating the model.

Repeating all these steps separately for each country multiplies costs, while re-using one single microsimulation framework for different countries or purposes can be a very *cost-effective* method of building new models.

3. THE STRUCTURE OF TAX-BENEFIT SYSTEMS

A general tax-benefit modelling framework will ideally be able to accommodate any existing or hypothetical tax-benefit system. In designing such a framework it is therefore essential to identify the principal elements of tax-benefit systems. In other words, it is necessary to find a suitable 'common denominator' of all (reasonably) possible structures. However, in general, there exists a trade-off between structure and flexibility: The modelling framework needs to provide the structure necessary for setting up a simulation model without limiting the breadth of tax-benefit systems that can be simulated.

In 'real world' tax-benefit systems, elementary policy rules are grouped together to form identifiable blocks such as 'instruments' (e.g., a tax credit), 'policies' (e.g., income tax), etc. In modelling a country's system, it is desirable to match the real system's hierarchy as closely as possible so that the logical representation provides a good intuitive equivalent of the original. From a model construction point of view it is desirable to try to generalise this representation as much as possible, so that most national systems can be described utilising the same structure.

Figure 1 shows the hierarchical structure that we use for a general tax-benefit modelling framework

and introduces the terminology used in the remainder of this paper. Each tax-benefit *system* is made up of individual *policies*. These are elementary collections of tax-benefit instruments. Examples for a policy are Income Tax, Social Insurance Contributions or Social Assistance Benefits. The *policy spine* is a list of policies indicating the sequence by which they are applied in the tax-benefit system. For example, if social insurance contributions are tax deductible, then the entry Social Insurance Contributions would have to appear *before* Income Tax because the model requires the amount of social insurance contributions as a prerequisite for calculating income tax; similarly, if social assistance benefits depend on after tax income, then the entry Social Assistance Benefits would have to appear *after* Income Tax since income taxes would be a necessary input for calculating Social Assistance. At the lowest level is the tax-benefit *module*, which performs the calculation of a certain part of the tax or benefit (e.g., a deduction, or applying a rate schedule to a tax base) on each fiscal unit. The modules represent the elementary building blocks of the tax-benefit system: only the modules contain actual tax-benefit rules. All other levels are merely necessary to structure these rules and to apply them in the appropriate sequence.

The tax-benefit structure that we have just described is linearly sequential. However in certain cases a decision must be made between a range of choices. For example a fiscal unit may be entitled to a range of different benefits and must choose one; or, as in the case of optional joint taxation, individuals may have to choose between being taxed individually and pooling income to form a joint tax base. Similarly, behavioural reactions to policy change are best modelled as an optimisation over a range of alternative decisions. The framework allows for such typologies by providing decision-rule elements. If required, these can be inserted anywhere along the policy spine to construct necessary 'branches' of a decision.

In terms of the actual computing environment, the MSM framework itself has been implemented as follows. In choosing the environment and programming language for the model framework, an effort has been made to ensure its longevity by not irrevocably attaching it to one specific computing environment. In addition, the aim has been to use software of a type that is familiar to potential users as well as easily available. However, care has been taken to avoid a rigidity, which would prevent future adaptations to other platforms such as UNIX.

The programming language used is C/C++ (Microsoft Visual C++). This facilitates efficiency in programming. However, the ability of C/C++ to write very streamlined and "direct" algorithms sometimes reduces the readability and transparency for less experienced users. As a rule, where trade-offs existed between transparency

and speed, we accepted decreases in the model's speed in return for improved readability and usability.

By using a method for database access (ODBC) which is available for all major relational database management systems, database systems other than the one used as a default can be used for data storage and management. Both the input micro-data and the model's micro-output (simulation results) can be stored in one of the widely used relational database systems (Oracle, Microsoft SQL, etc.) Microsoft Access is used as the default. Input and output data are stored in two separate databases. In this way, the input micro-data can remain "read-only". However, the relational data structure makes it possible to combine the physically separate input and output data into one logical table to analyse the impact of a tax-benefit system in relation to all sorts of characteristics (age, household size, etc.).

All parameter lists are stored as spreadsheet tables and can be read and manipulated with any spreadsheet software (e.g. Microsoft Excel).

4. GENERALISATION & PARAMETERISATION

In order to accommodate the numerous requirements and different uses outlined in section 2 and to 'fill' the individual boxes of the hierarchy described in section 3 with actual tax-benefit algorithms, the modelling framework will necessarily need to be quite generalised. The degree of generalisation relates to the degree to which a model is 'parameterised' so that model code can be used for different purposes without re-coding. For instance, a 'tax-schedule' module that is programmed in a way that works with any number of tax bands and any set of tax-rates can be re-used for modelling the income tax system of many different countries. While generalising as much as possible makes the model more flexible, it also has the effect of making it more difficult to develop and also potentially less transparent, conceptually and computationally more complex (and, hence, slower) than a similar model which is built for a narrow and *a priori* clearly defined set of applications.

As briefly mentioned above, designing any MSM framework is a very resource intensive task. In a survey by Mot (1992), it was found that static *national* models generally took 2 to 3 man-years to develop. More sophisticated models, such as the TRIM2 model in the USA, took much longer. Although more costly to build initially, a *generalised* model is less costly in the long run if the framework can be re-used for a multitude of purposes. In addition, the robustness and reliability of a modelling framework will be positively related to the number of users and uses. As a result, a generalised modelling platform will 'mature' more quickly than purpose-built model frameworks with a more narrowly defined scope. A generalised multi-purpose framework will also facilitate communication and co-operation

between researchers and reduce training costs as many people will share similar experiences and problems.

The use of parameterisation aids all the primary objectives of a generalised framework outlined in section 2. Modelling flexibility can be achieved by enabling the user to make adaptations and definitions without changing the program code itself, making changes instead to certain sets of parameters. As a result, the specification of policy reform scenarios becomes easier. The parameterisation of tax-benefit algorithm also ensures that their operation remains transparent and adaptable which in turn make the model more robust. Similarly, model maintenance is considerably simplified if all one needs to do to update policy rules is to change the relevant parameters (such as new amounts, rates or definitions). Parameterising common model elements which can be used in many different contexts also aids consistency and transparency. An example is the generalised social benefit routines described below.

Parameterisation also facilitates experiments with different parameter values. For example, one may wish to analyse how including unemployment benefits in the income tax base, changing the definition of the fiscal unit (e.g., changing the maximum age at which a person is considered a 'child' for the purpose of computing a certain benefit), or altering the number of tax brackets affect the distribution of disposable income. In the present framework, all such changes can be specified by setting appropriate parameters in the parameter lists. The actual tax-benefit algorithms are coded as functions of these externally defined parameters and will not normally have to be accessed by the model user.

The key elements of the tax-benefit framework that are parameterised are:
1. *Modules*, the primary building blocks of the model. Components to be parameterised include the definition of parameters directly related to the tax-benefit algorithm relevant for each module (e.g., rates, bands thresholds, type of income concepts, fiscal units).
2. *Policies* and *Policy Spine*, the structuring mechanism within the framework. The parameters relate to the types of module/policy (i.e., which modules make up a policy and which policies make up the tax-benefit system) as well as their order.
3. The definition of the *fiscal units* relevant for an instrument (e.g., who belongs to a 'family' receiving the instrument, who belong to a 'couple' whose income is taxed jointly, who counts as a 'child' for the purpose of computing child benefits).
4. The definition of *sharing rules* within the unit (i.e., which unit member receives what part of a benefit and how are tax burdens shared between members of the tax unit).
5. The definition of *aggregate income concepts* that combine income variables used either by

an instrument (e.g., 'taxable income' such as market incomes plus benefits minus deductions and allowances) or as an output of the model (e.g., 'disposable income').
6. The set of *variables* to be used in the framework as well as their characteristics such as whether they are to be simulated (e.g., taxes) or read from the data (e.g., employment income) or whether they are monetary variables.
7. '*Uprating' factors* for each monetary variable, used for updating purposes. If the data was collected in 1996 and the policy we wish to examine is for 1998, then we need to alter the data to bring all monetary variables forward to 1998 (accounting, e.g., for inflation, earnings growth, etc.).
8. *Output functions*, including the variables to be written to the output file, as well as the types of summary statistics required as output.

In the following section we describe the implementation of these key concepts and elements of MSMs in more detail.

5. IMPLEMENTATION OF KEY CONCEPTS FOR TAX-BENEFIT MODELLING

In the previous section, we have alluded to a number of different concepts such as 'modules'. 'fiscal units', 'sharing rules', 'income concepts' and 'common routines'. In this section, we shall discuss the actual implementation of these elements in the MSM framework.

5.1 Modules
The concept of modules as distinct building-blocks of the simulation has special advantages. By using the same building blocks for different tax-benefit systems, one can build up a large 'library' of national specific modules. It is therefore possible to select modules from this library when either designing a new system or examining the effect of introducing aspects of other systems in a given country. The flexible order of modules and the high degree of parameterisation ensure that the same modules can be used for a multitude of different purposes.

In terms of the actual implementation, there a number of desirable features. The structure of the module should be a function with clearly defined inputs and outputs. The body of the function should look similar for each tax-benefit instrument. Every section should be clearly labelled and documented so that users wishing to adapt an existing instrument would readily see where changes have to be made while those wishing to implement a new instrument would only have to fill in the blank spaces. It should be possible to freely define intuitive variable names used in the algorithm to make interpretation as straightforward as possible.

Figure 2 describes the general structure of modules as implemented in the framework. The
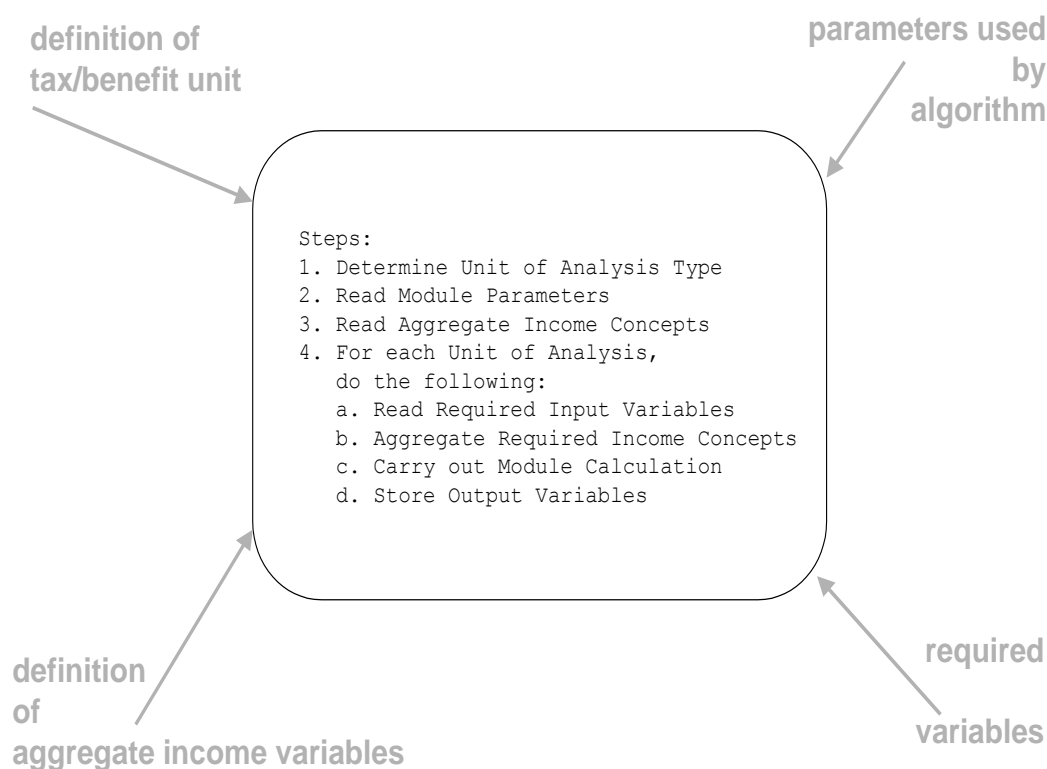
definition of
tax/benefit unit

parameters used
by
algorithm

```
Steps:
1. Determine Unit of Analysis Type
2. Read Module Parameters
3. Read Aggregate Income Concepts
4. For each Unit of Analysis,
   do the following:
   a. Read Required Input Variables
   b. Aggregate Required Income Concepts
   c. Carry out Module Calculation
   d. Store Output Variables
```

definition
of
aggregate income variables

required

variables

Figure 2 Module description

inward arrows define the set of input variables. Before the actual tax-benefit algorithm coded into the module is initiated, each module performs several steps. The first step involves determination of the relevant fiscal unit such as individual household or family etc. that the instrument applies to. The next step involves reading the parameters such as rates, bands thresholds, age limits etc. used by the algorithm. Other parameters include the definitions of aggregate income concepts used within the module such as earnings, total benefits and tax-base. Once these parameters have been read, the actual calculations take place.

To minimise the scope for errors and interference with other parts of the model, each module should only contain those parts of the program code, which are absolutely necessary for specifying the algorithm. Everything else should be "hidden" from the user.[5] Applying this design philosophy ensures that individual modules can be developed independently and thus the operation of one module does not interact with the operation of other modules. This improves the robustness of model, so that errors in one part of the code do not influence other parts. Once a module has been thoroughly tested and is found to work, it can be added to the system as a whole. This is a method also employed to varying degrees by the US model TRIM2 (Mot, 1992) and the Institute for Fiscal Studies model, TAXBEN (Giles and McCrae, 1995). Equally important, a strictly modular design simplifies the maintenance of a model considerably. It ensures that changes, which inevitably have to be made during the lifetime of any tax-benefit model, do not have unexpected

side effects on other modules and enables separate parts of a MSM to be developed independently by different members of a team. For the development of EUROMOD, this has been a critical feature since the implementation of 19 tax-benefit systems cannot be completed by one person alone.

In order to support safe and efficient implementation of new instruments, the MSM framework provides a large number of frequently used standard functions. These are routines that perform operations or determine characteristics which are relevant for the calculation of a large number of policy instruments (e.g., NUMBERCHILDRENINTAXUNIT, ISMARRIED, ISLONE PARENT). This approach ensures a consistent interpretation of variable values (especially categorical variables, such as marital status) across all modules, considerably simplifies the maintenance of the model and means that users do not have to access variables in the microdata directly.

Table 1 illustrates typical module parameter sheets. Each module starts with the name of the module (CO_IT_MAIN_TFA; CO_IT_ SCHEDULE) followed by a number of parameters. Parameters common to each module are TAX_UNIT and SWITCH. The former specifies the fiscal unit relevant for this module while the latter determines whether or not this module should be included in the calculations. All other parameters are module-specific (i.e., they depend on the specific tax-benefit algorithm coded into the module).

In addition to modules that have been designed

Table 1 Example Policy Parameter Sheet

param_no	param_name	Period	first_system	UK_1998	end_system
	first_module			**Income Tax**	
	...				
	co_it_main_tfa				
1	tfa	y		4195	
2	age_limit1			0	
3	age_limit2			64	
4	TAX_UNIT			individual	
5	SWITCH			1	
	co_it_schedule				
1	nbands			3	
2	tax_band1	y		4300	
3	tax_band2	y		27100	
4	tax_rate1			0.2	
5	tax_rate2			0.23	
6	tax_rate3			0.4	
7	TaxableY_il			TaxBase	
8	TAX_UNIT			individual	
9	SWITCH			1	
	...				
	end_module				

for a specific purpose in a specific country the framework also provides a large number of common modules, which were designed without any single country or use in mind. Instead they can be used for many different purposes. Examples are schedules where the number of rates or other parameters is flexible and where the income base to which the schedule is to be applied can be freely defined (e.g., CO_IT_SCHEDULE in Table 1). All these parameters can be specified in parameter sheets, which means that in many cases, even very complicated instruments can be implemented without any need for programming. Apart from the considerable amount of time and effort that can be saved by re-using already existing building blocks, there is, again, the added advantage that these general modules have already been thoroughly tested. One can therefore be confident that the risk of programming errors is minimal.

Because for many countries, the sub-components of social benefits can be classified in a similar manner across countries, it is possible to classify benefits into a number of common modules, resulting in one of the most powerful set of common modules in the framework. Eligibility is determined first. If a unit is eligible for the benefit then their 'means' (i.e., the income that is set against a benefit) are calculated.[6] Next we specify the 'equivalence scale' for determining the benefit amount as a function of characteristics of the fiscal unit (such as age, number of people in the family and number of children). By imposing a common structure on seemingly often very different benefits, it also allows for the relative generosity of benefits for different family types to be easily compared across countries without having to adjust for currency differences. Finally,

the majority of all benefits in western countries can be classified as the base amount times the equivalence scale minus means times a withdrawal rate (r).

Both eligibility and equivalence scale modules contain large numbers of different types of parameters to permit modelling of many different types of benefits and reforms. A list of parameters that exist in the framework and can be used for specifying eligibility conditions and equivalence scales is provided in appendix 2 of Immervoll and O'Donoghue (2001a). In total, one can chose between more than 1000 possible parameters. Because such a large number of parameters would be extremely difficult to manage, we allow the user to select only the parameters they wish to use, ignoring the rest. The parameters can therefore be seen as a library of possible conditions to determine eligibility and equivalence scales. All eligibility conditions can be combined using a combination of logical AND, OR and NOT operators as described in the appendix of Immervoll and O'Donoghue (2001a). In constructing EUROMOD, we were able to implement almost all benefit instruments that existed in the EU using the common modules described here.

5.2 Policies and Policy Spine
In this section we describe how *policies* and the *policy spine* operate in the generalised framework. Different policies can only communicate with each in terms of well-defined output variables as specified by the model user. For example, if the only output of the Social Insurance Contribution policy is a variable called SIC then the only way this policy can influence the calculations of other policies (e.g., Income Tax) is via this variable.

For example, if social insurance contributions are tax-deductible the variable SIC would be subtracted from the tax base during the implementation of the Income Tax policy. Policies therefore improve robustness by preventing any unintended interactions.

A significant feature of the MSM framework is that the order in which both policies within the spine and modules within policies are simulated can be altered by the user without recoding the model. For example if one decides to make child benefits taxable, one would place child benefits before income tax in the spine (and include them in the definition of the tax-base), while if child benefits were to be means tested on post tax income, one would put child benefits after income taxes in the spine. Similarly, the sequence of the modules contained in the policy determines the sequence of module calculations. For instance, in the case shown in Table 1, the main tax allowance is computed before the income tax schedule is applied to the TAXBASE.[7] The sequence can be changed by simply moving around the parameter blocks in the parameter sheet.

5.3 Definition of Fiscal Units
Tax-benefit rules relate to certain fiscal units, i.e., the person(s) on which the tax-benefit rules are to be performed (e.g., the persons over whom taxable incomes are to be aggregated in order to determine total taxable income in a joint tax system). In the present framework, each module must contain the name of the *type* of fiscal unit on which the tax-benefit algorithm is to be performed (e.g., INCTAX_UNIT). Fiscal unit types themselves can be defined in a separate parameter sheet.

In the simplest case, the fiscal unit type is either the largest identifiable unit in the micro-data (usually the 'household') or the smallest (the individual). If it is neither then one has to define exactly which members of the largest unit (household) belong to the same unit as the 'head' of the fiscal unit. Possible choices are Cohabiting Partner, Married Partner, Child and Dependent Parent. For the latter two, a powerful set of conditions is available for defining what constitutes a 'child' or a 'dependent parent' (including age limits, income limits, conditions relating to marital-, labour market-, or education status). All of these conditions can again be combined with logical AND, OR and NOT operators. A pseudo-code of the routine used to assign people to fiscal units is described in Appendix 1 of Immervoll and O'Donoghue (2001a).

In each household, there may be one or more instances of a fiscal unit *type*. For each fiscal unit type, each person in the household receives a number indicating the fiscal unit (of this type) they belong to. Using the conditions mentioned above, it is possible to decide for each person whether or not they are member of a specific fiscal unit.[8] A fiscal unit can be fully or partly occupied so that if the fiscal unit type is, for example, 'married couple' then one person living in a one-person household can be allocated to the fiscal unit of type 'married couple' even though there is no spouse present. Persons who are not assigned to a fiscal unit together with other persons form their own fiscal unit.

5.4 Sharing benefits and tax burdens within the Fiscal Unit
By default, the outcome of all tax-benefit instruments is assigned to the head of fiscal unit. However frequently it is desirable to be able to use other incidence assumptions. In order to do this, it is necessary to provide information about assumed sharing arrangements. The framework supports a number of different assumptions. As currently implemented, it is possible to share amongst:
- Adults/children;
- Economically active/inactive persons;
- Part-time/full-time workers;
- Male/female head of unit

The instrument to be shared can be divided equally amongst all those to whom the instrument is to be shared, or divided in proportion to the level of a particular income amount held by each individual. Allowing such explicit definitions of intra-unit assignments of taxes/benefits, it becomes possible in principle to analyse simulation results at any level of analysis (e.g., gender specific), rather than just at the household level.

5.5 Income Concepts
Income concepts used in the tax-benefit algorithms (e.g., taxable income and 'means') or as output of the model (e.g., disposable income) can be defined in terms of all monetary variables available in the model, whether contained in the micro-data or simulated by the tax-benefit model. Each income concept is defined in terms of a vector of numbers between −1 and +1. The size of the vector is equal to the number of monetary variables in the model. For each of the variables, the number in the vector indicates what fraction of this monetary variable is part of the income concept. For example, if 'mortgage interest payments' are deductible from taxable income then the 'taxable income' vector would contain a '−1' entry for the 'mortgage interest payments' variable.

5.6 Data Manipulation
In addition to parameters related to the tax-benefit algorithms *per se*, a number of parameters included in the modelling framework relate to the micro-data on which the tax-benefit system is to be simulated. One of the desirable features of a microsimulation modelling framework is that it should be possible to add new variables with ease. To this end, all variables used in the model are specified in a list containing the variable names and additional information such as whether the variable relates to individuals or to households as a whole, and whether or not it is a monetary variable. Once specified in this way, the model

automatically carries out all the procedures necessary to make the variable useable by the model (i.e., as input or output of a tax-benefit calculation).

For cases where certain variables are not available in the micro-data underlying a simulation, default values can be specified. This is especially important in a multi-country context, where one may want to simulate the effects of introducing a tax-benefit instrument from country A in country B. If the tax-benefit rules of this instrument require a variable which is not available in country B's micro-data, then one can specify appropriate default-values for this variable. Default values can be specified either directly (i.e., by specifying an actual value) or by referring to a variable that *is* available in the micro-data and is considered a good approximation of the missing variable. For example, in a situation where the tax-benefit rules of country A require information on whether someone is a civil servant and where there is no civil servant variable in country B micro-data, one can specify that the variable 'public sector' available in country B data should be used as a proxy for 'civil servant'.

5.7 Updating
Frequently the data available for microsimulation are not from the same year as the year to which the policy scenario of interest relates. This is because tax-benefit policy changes most years, while data are often only collected infrequently. To still be as 'representative' of the population as possible, adjustments are necessary. The first of these relates to adjustment of the weights in the data. Because aspects of the population may have changed between data collection and the year of analysis, it may be desirable to adjust the weights in the data to account for these changes. Examples of population changes include the level of unemployment, the number of households with children and the age distribution. It should be noted that extreme care needs to be exercised when doing this. 'Correcting' the data by adjusting weights in relation to many different dimensions can cause anomalies (such as very big weights for certain household types), because re-weighting in relation to any set of characteristics (e.g., the level of unemployment) will invariably distort other dimensions for which the weights have originally been designed for (e.g., non-response, regions).

In addition, the values of monetary variables will have changed due to price changes and real increases between 'data year' and 'policy year'. This aspect has been parameterised in the MSM framework, where uprating factors can be specified for each monetary variable. Different incomes may increase at different rates and since these rates may themselves differ for different groups (for example employment income may increase at a different rate for males/females, civil servants etc.), we allow for differential uprating. A potential use of the updating mechanism is for short-term forecasting of fiscal variables, where

the underlying data is adjusted to match expected changes in the population and incomes. Data thus adjusted can then be used as input into the tax-benefit model to explore projected aggregate revenue/costs and or distributional features.

5.8 Output
The principal output of a simulation run is a micro output file that can output any variable for any fiscal unit in the model. The micro output file can then be used with any statistical package for performing more elaborate analyses. An important feature of the output routine is that it can be integrated into the policy spine of the tax-benefit system just like any other 'policy'. This means that it is possible to produce output *at any stage* of the tax-benefit calculation and thus trace variables of interest. Users can, for example, specify that they want output to be generated both before and after the Social Assistance policy instrument. By comparing different outputs, one can then easily observe the differential impact of one individual policy or a set of policies (whether for analytical or model validation purposes). Of course, the typical position of the output element will be at the very end of the policy spine, writing simulation results for the tax-benefit system as a whole.

Even though any desired statistical package can be used to analyse these micro-level outputs it is, for a number of reasons, desirable to have most of this analytical capability available within the model framework. Keeping track of numerous large micro-output files for many different simulation runs can be difficult and a source for errors. This is the case for any tax-benefit model since it is often necessary to formulate a large number of policy scenarios in order to explore the research question at hand. For a multi-country model, the number of output files is potentially much larger. In addition, the total sample size of the micro-data underlying a multi-country MSM can be very large. In the case of EUROMOD, these data represent more than 100,000 households containing more than a quarter of a million people. Any multi-country micro-output will therefore be of a similar size which may exceed the relevant limits of some commercially available software tools. Most importantly, analysing micro-output can be very time consuming. Many different analyses are possible and each of them entails a set of assumptions and definition which needs to be decided upon. As a result, it is convenient to have a 'standard output' which can reliably perform most of the desired analyses while keeping all the related choices and assumptions as transparent and accessible to the user as possible. In this way, it is possible to ensure consistent output across uses, users and countries. Users can rely on the 'standard output' routine to have been tested and to produce correct calculations that are robust and consistent across different applications.

A standard output routine has, therefore, been embedded into the MSM framework. As

implemented it can also be run separately and can thus be used as an analytical tool for analysing any micro-data file, whether generated by a tax-benefit model or not. The standard output routine is designed to:

- provide statistics and summary indicators that are accepted standards among researchers and policy analysts and that can be used in a consistent way across different countries and uses of the model;
- permit users to analyse the sensitivity of the various indicators by allowing them to vary underlying concepts and definitions such as exchange rates, poverty lines, equivalence scales, etc.;
- mirror the flexibility of the tax-benefit simulation framework by not imposing any a priori definition of concepts such as disposable income, a 'child', etc.;
- be able to handle the very large amounts of data resulting from the simulation of policy instruments for all households contained in micro-datasets of several or all EU countries;
- provide a user interface that is similar to that in other parts of EUROMOD;
- attach a comprehensive description to the numerical output which clearly shows the kinds of choices made by the user of the output program. Given the multitude of possible definitions and concepts such 'labelling' is essential to ensure that the numbers produced by the output program are interpreted in an appropriate way;
- be computationally reliable and robust.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper describes the rationale for microsimulation tax-benefit modelling and the demands placed on the method by users. We note the high cost of developing microsimulation models in different countries and argue for the need to control these costs. We present a generalised microsimulation framework that, if adopted, provides substantial economies of scale in the design of microsimulation models. Although the time taken to construct this general framework has been considerable, the subsequent economies of scale that result have already become evident as the framework and its components have successfully been used to implement an integrated European tax-benefit model comprising the tax-benefit systems of 19 EU member countries. This implementation has taken three years whereas a separate development of individual country specific MSMs would have taken much longer[9].

The principal design feature of the generalised framework is the extent to which routines and operations have been generalised and parameterised and are thus re-usable for different purposes. Examples not typically found in national specific tax-benefit microsimulation models include the consistent parameterisation of:

- The fiscal unit of analysis;
- The order in which instruments are simulated;
- Income concepts; and
- The input database and related operations such as data updating, etc.

The use of encapsulated policy components makes resulting tax-benefit models flexible and robust. It also allows users to focus on those parts of the tax-benefit system which are of interest for the research question at hand while not having to worry about computational details of the modelling framework as a whole.

A frequent criticism of static microsimulation modelling is that it only measures the day-after effect while some reforms may initiate relevant behavioural responses such as reduced work effort. While incorporating behavioural changes can be a worthwhile exercise, its usefulness depends on the questions to be addressed. For example, for the purpose of understanding the interactions between different tax-benefit instruments a model, which does not mix immediate effects with longer-term behavioural dimensions, will often be preferable. Nevertheless, because of the framework, future expansions of the model to incorporate behavioural response are feasible without radical redesign. For example Colombino et al. (2008) introduced a module that called the tax-benefit system in EUROMOD to generate a budget set required to estimate a labour supply model in a number of EU countries.

Notes

[1] Although it can, for some purposes, be useful to analyse the effects of policy on hypothetical populations, these models typically build on micro databases drawn from either household surveys or administrative register information. Using such data on actual populations, MSMs can be used as tools to analyse 'real-world' effects of social and fiscal policy.

[2] For example Albuquerque et al. (2001) have used this framework to examine the efficiency of social protection measures in Southern European countries.

[3] In Immervoll and O'Donoghue (2001a) we have used EUROMOD to study interactions of welfare benefits, taxation and work incentives in four European countries.

[4] Hierarchal in this case refers to the hierarchy of complexity, where the user interacts less with the more complex features and more with the less complex.

[5] Such modularisation and encapsulation are well known programming principles (see, for instance, Wiener and Pinson, 1988).

[6] In some instruments proportions of certain income sources such as earnings may be disregarded.

[7] The reason being that the tax free allowance is subtracted from the tax base.

[8] A person can be a member of more than one fiscal unit simultaneously (e.g., 'individual', 'married couple' and 'household') but he/she

can only be a member of one fiscal unit of a given fiscal unit type (e.g., if two married couples live in the same household, each person is only allowed to be a member of one 'married couple' unit).

[9] To view working papers and other publications that use EUROMOD or to view related projects, see the EUROMOD website: http://www.iser.essex.ac.uk/msu/emod/

REFERENCES

Albuquerque J L, Bargain O, Baldini M *et al*., (2001) 'The impact of means tested assistance in Southern Europe', EUROMOD Working Paper 6/01, Department of Applied Economics, University of Cambridge.

Atkinson A B, Bourguignon F and Chiappori P-A (1988) 'What do we learn about tax reform from international comparisons? France and Britain', *European Economic Review*, 32, 343-352.

Atkinson A B, Bourguignon F , O'Donoghue C *et al*. (2002) 'Microsimulation of social policy in the European Union: Case Study of a European Minimum Pension', *Economica*, 69(274), 229-243.

Bourguignon F, O'Donoghue C, Sastre-Descals J *et al*. (1997) 'Eur3: a prototype European tax-benefit model', Microsimulation Unit Discussion Paper No. 9703, Department of Applied Economics, Cambridge.

Callan T, O'Donoghue C and O'Neill C (1996) *Simulating welfare and income tax changes: the ESRI tax-benefit model*, Dublin: The Economic and Social Research Institute.

Callan T and Sutherland H (1997) 'The impact of comparable policies in European countries: microsimulation approaches*, European Economic Review*, 41(3-5), 627-633.

Citro C F and Hanushek E A (*eds*.) (1991a) *The uses of microsimulation modelling, Volume 1, Review and Recommendations*, Washington: National Academy Press.

Citro C F and Hanushek E A (*eds*.) (1991b) *The uses of microsimulation modelling, Volume 2, Technical Papers*, Washington: National Academy Press.

Colombino U, Locatelli M, Narazani E *et al.* (2008) 'Behavioural and welfare effects of basic income policies: a simulation for European countries', CHILD – Centre for Household, Income, Labour and Demographic Economics Working Papers, wp03_08, Dept. of Economics, University of Turin. (Available from http://www.child-centre.it/)

Giles C and McCrae J (1995) 'The IFS micro-simulation tax and benefit model', IFS Working Paper W95/19, Institute for Fiscal Studies, London.

Hancock R (1997) 'Computing strategy for a European tax-benefit model', Microsimulation Unit Discussion Paper MU9704, Department of Applied Economics, University of Cambridge.

Hoshka P (1986) 'Requisite research on methods and tools for microanalytic simulation models', in Orcutt G, J Merz and H Quinke (eds.) *Microanalytic Simulation Models to Support Social and Financial Policy*. Amsterdam: North-Holland.

Immervoll H (2000) 'Fiscal drag - an automatic stabiliser?', Paper presented at the 56th Congress of the International Institute of Public Finance, Seville, August 28-31.

Immervoll H and O'Donoghue C (2001a) 'Welfare benefits and work incentives: the distribution of net replacement rates in Europe', EUROMOD Working Paper 4/01, Department of Applied Economics, University of Cambridge.

Immervoll H and O'Donoghue C (2001b) 'Net to Gross', EUROMOD Working Paper 1/01, Department of Applied Economics, University of Cambridge.

Immervoll H, O'Donoghue C and Sutherland H (1999) 'An introduction to EUROMOD", EUROMOD Working Paper 0/99, Department of Applied Economics, University of Cambridge.

McCrae J (1999) 'The development and uses of tax and benefit simulation models', *Brazilian Electronic Journal of Economics*, Vol 2 No 1.

Merz J (1991) 'Microsimulation - a survey of principles, developments and applications', *International Journal of Forecasting*, 7(1), 77-104.

Mot E S (1992) *Survey of microsimulation models*, Social Security Research Committee, The Hague: VUGA.

O'Donoghue C and Sutherland H (1999) 'For richer, for poorer?: The treatment of marriage and the family in European income tax systems', *Cambridge Journal of Economics*, 23(5), 565-598.

Piachaud D and Sutherland H (2000) 'How effective is the British government's attempt to reduce child poverty?', CASE paper 38, CASE, London School of Economics.

Redmond G, Sutherland H and Wilson M (1998) *The arithmetic of tax and social security reform: a user's guide to microsimulation methods and analysis*, Cambridge: CUP

Sutherland H (1991) 'Constructing a tax-benefit model: what advice can one give?', *Review of Income and Wealth*, 37(2), 199-219.

Sutherland H (1995) 'Static microsimulation models in Europe: A survey', Microsimulation Unit Discussion Paper, MU9503, Department of Applied Economics, Cambridge.

Wiener R S and Pinson L J (1988) *An introduction to Object-Oriented Programming and C++*. New York: Addison-Wesley Publishing.

Weinberg DH (1999) 'Income data collection in international household surveys', Paper presented at the 3rd Meeting of the Canberra Group in Ottawa, Canada, June 7-9.