# Stock Flow Diagram Making with Incomplete Information about Time Properties of Variables

**Yutaka TAKAHASHI**
School of Commerce, Senshu University
2-1-1, Higashimita, Tama, Kawasaki, Kanagawa, Japan
Postal code: 214-8580
Telephone: +81-44-900-7988 Facsimile: +81-44-900-7849
takahasi@isc.senshu-u.ac.jp

***Abstract***

*Models for numerical simulations should be described in a coherent style. They are expected to have consistencies at the causal dependency level. However, System Dynamics causal loop diagrams can have inconsistencies. This diagram style's arrows, concerning flow and stock relationships, can have the opposite direction of stock flow diagrams which can numerically simulate models. These inconsistencies can cause inappropriate qualitative simulations so that it is sometimes recommended to use stock flow diagrams instead of causal loop diagrams even for qualitative simulations. However, causal loop diagrams have merits in their use. Causal loop diagrams are intuitively easy to draw and read. If causal loop diagrams are given information about each variable's dynamic property, they can be changed to stock flow diagrams and simulation models can be generated. This paper suggests how to use causal loop diagrams as a starting point in numerical simulation research.*

Keywords: stock flow diagram, natural language, automatic modelling

## 1. Introduction

System Dynamics modelling is now assisted by computer software. Most System Dynamics software has a GUI so that model builders can obtain equations from their

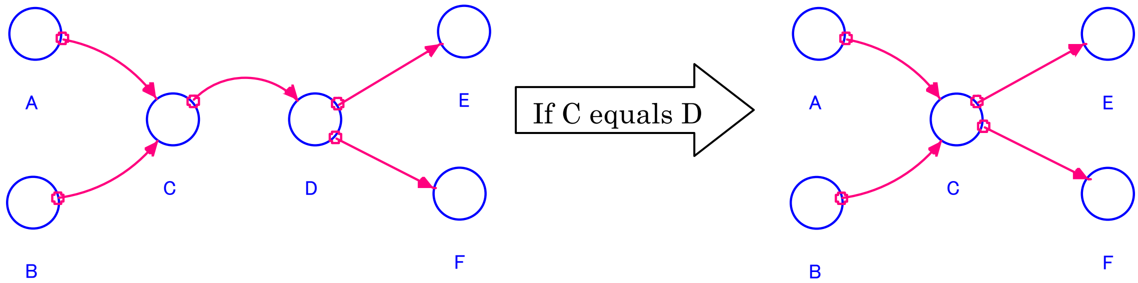


Figure 1: Most model builders eliminate redundancies

mental models which are not defined completely, and the software checks consistencies of model structures.

Nevertheless, some people find difficulties in making or using System Dynamics. There are several possible reasons for this. The most accepted one is that people are not used to dealing with dynamic concepts. General people are not always aware of dynamic behavior (Sweeney and Sterman, 2000; Ossimitz, 2002; Kainz and Ossimitz, 2002). Therefore, it is possible that not only seeing or reading but also writing dynamics is difficult for them.

From the viewpoint of Sweeney and Sterman (2000), several approaches which assist model builders to understand and deal with dynamic structures or variables appropriately have been suggested. Richmond (1992) gives some metaphors in order to make correspondences between parts of speech in natural language and icons (varieties of variables) in stock flow diagrams. This way of thinking is useful, but he presented at a heuristic level. Therefore, people need some “on the job training” in order to effectively take advantage of this idea.

This idea was extended in the research by Takahashi (2005), which defines correspondences in every combination of variables in stock flow diagrams. Model builders can directly obtain stock flow diagrams by translation from natural language which is used in order to describe mental models.

Although translation correspondences are indicated, their applications to generate stock flow diagrams still require us to be experienced in System Dynamics modelling. The reason for this is that relationships between any two sentences are not intuitively understood. Because of ambiguities of natural language, model builders can give multiple names for one quantity. Indeed, relationships between any two sentences are generated by the existences of common variables in those sentences. Therefore, it is possible that model builders find no relationships between two sentences which in fact do have relationships. Direct translation from natural language to stock flow diagrams has such a frailty caused not by the process of translation but by ambiguities of natural language.

Classic causal loop diagrams have less possibility of causing this kind of trouble. This diagram style is widespread after publication of Senge (1990) for drawing one's issues or situations. In the process of drawing causal loop diagrams, model builders unconsciously choose words for their own causal loop diagrams and eliminate redundancies of elements. Variables which have the same meaning can exist in a causal diagram. Nevertheless, such redundant items' existence only causes visual complexities. Model builders can find such redundancies so that they can reorganize the diagram (figure 1).

However, causal loop diagrams have a weak point in that diagrams are not directly connected to numerical simulation models. One is the meaning of polarities pointed by Richardson (1997). Richardson (1997) extends the definitions of polarity and solved the problem. On the other hand, Takahashi (2005) indicates that the directions of arrows between a stock and flow variables in causal loop diagrams are the opposite to those of stock flow diagrams, especially about outflow variables and their stock variables. It causes the requirement of additional consideration of dependency relationships between any two variables in causal loop diagrams when they are examined numerically.

Although causal loop diagrams have such a weak point, this diagram style still has a merit in its use: the figure of diagrams themselves. In a causal loop diagram style, all parts are consistently connected. Model builders who make causal loop style have no worries about how to connect the sentences which consist of their own mental models. Therefore, if elimination of causal loop diagrams' inconsistency with numerical models, or in other words, addition of dynamic characteristics to causal loop diagrams is achieved, model builders can make dynamic simulation models starting from causal loop diagrams.

This paper presents a method for addition of dynamic characteristics to causal loop diagrams with a less workload on model builders than an ordinary modeling process.

## 2. Methods

It is necessary to give the time properties (stock, flow, or auxiliary variable) to causal loop diagrams to make information which is required when numerical simulations are employed. This means that each variable's characteristics which can be found in causal loop diagrams need to be understood. If these characteristics are defined, irrespective of model builders, causal loops can be dealt with as stock flow diagrams. This means that causal loop diagrams can become a starting point for numerical simulations.

In order to discover the necessary information for numerical simulations, the kind of information that is contained in stock flow diagrams which are employed in numerical simulations is examined.

Firstly, a method of indicating variables' time properties is examined. Stock flow diagrams hold dynamic structures which are expressed using various icons and arrows. The usage of these icons and arrows are strictly defined and directly connected to differential equation structures. On the other hand, causal loop diagrams do not have such varieties of icons because they deal with all variables equally in the time property. Therefore, this characteristic of stock flow diagrams cannot be introduced into causal loop diagrams.

Secondly, a method how to indicate relationships between variables is examined. All variables are connected by arrows in stock flow diagrams. This characteristic can be seen in causal loop diagrams. Therefore, if information regarding how each time property variable is connected to other variables is examined, this information can be used in generating numerical simulation tools from causal loop diagrams.

Then, characteristics of inter-variables connection by each time property are examined.

Stock variables are connected only to their own flow variables. In other words, such flow variables must be finite differences of the connected stock variables. Stock variables cannot have any connection started from any other variables, such as another stock variable's flow variables or auxiliary variables. Moreover, stock variables cannot be auxiliary variables at the same time.

This characteristic can be expressed as below using a predicate logic formula. This predicate logic formula is used to obtain computer programs which decide variables' time properties automatically without coding costs. In this formula, the predicate "*P(X, Y)*" means "*X* has a time property of *Y*," and predicate "*L(X, Y)*" means "A connection from *X* to *Y* exists," the operator "$\neg$" means logical "not." Capital letters are logic formula variables which contain elements in causal loop diagrams.

$$L(W, X), P(W, flow), \neg P(X, auxiliary) \rightarrow \forall X\,(P(X, stock)) \quad (1)$$

Next, flow variables have connections from any variable. This is not valuable information. Moreover, a flow variable has connections to any variables including stock variables whose change is defined in the flow variable. Of course, flow variables cannot be auxiliary variables simultaneously.

Using the same definition of predicates, flow variables can be defined in a predicate logic formula as below.

$$L(X, Y), P(Y, stock), \neg P(X, auxiliary) \rightarrow \forall X\,(P(X, flow)) \quad (2)$$

Finally, like flow variables, auxiliary variables have connections from any variable. However, auxiliary variables cannot directly connect to stock variables. Of course, auxiliary variables must not have another time property at the same time. It is the same as other time property variables.

Using the same definition of predicates, auxiliary variables can be defined in a predicate logic formula as below.

$$L(W,X), \neg P(W,stock), \neg((P(X,stock)) \cup (P(X,flow))) \rightarrow \forall X(P(X,auxiliary)) \quad (3)$$

These formulae have circular definitions (figure 2). According to the definition of relationships between elements in System Dynamics models, a time property of one variable is defined by relationships to other variables. However, the existence of circular relationships does not allow to do automatic reasoning of time properties. Therefore, in order to eliminate these circulations, a choice or a definition of variables whose time properties are common requires setting these variables' time properties in advance of the definition of other variables' time properties.

In small models, to give an appropriate time property to each variable might not be difficult. However, when models are relatively complicated or multiple people are involved in modelling work, some guidelines are needed. In such situations, the correspondences suggested in Takahashi (2005) are useful. This indicates correspondences between natural language sentences and stock flow diagrams. Using it, model builders can find stock variables (or other time property variables) in

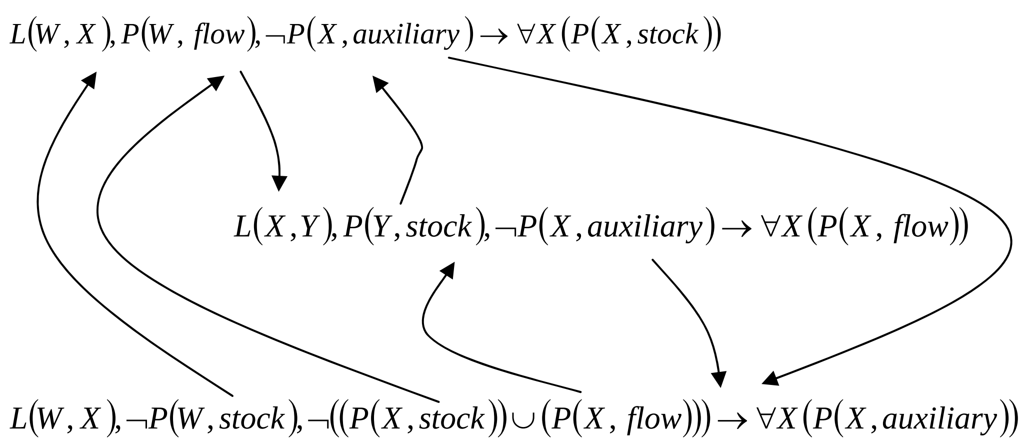


Figure 2: Circular definition

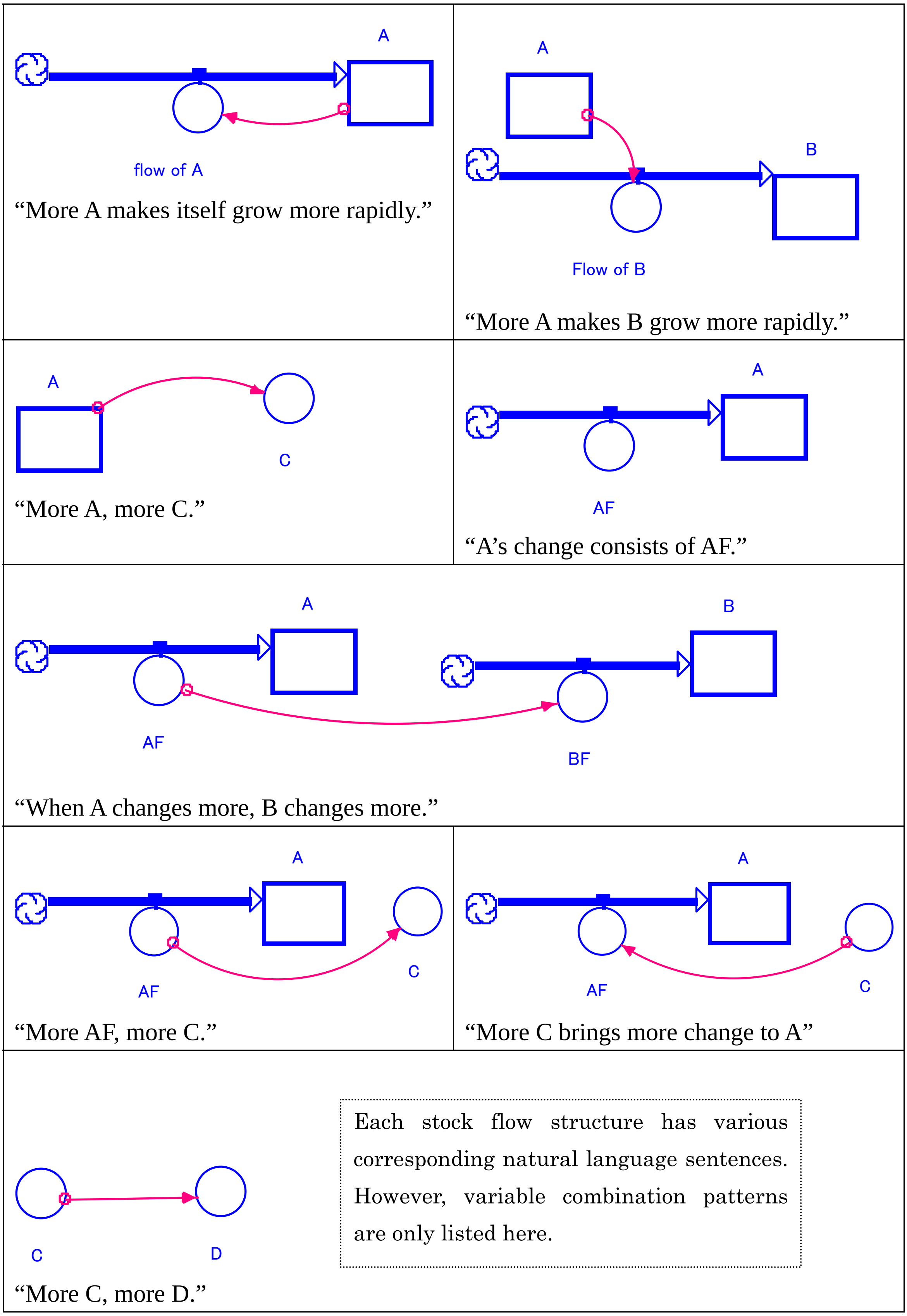


Figure 3: Translation from natural language to stock flow diagrams

their mental models or causal loop diagrams. Organized correspondences between a stock flow diagrams and natural language sentences are shown in figure 3.

If model builders decide to choose stock variables in their causal loop models, they can use the predicate logic formulae (2) and (3) in order to define other variables' time properties. This process allows no exception. Therefore, all model builders can obtain the same result when they hold the same choices of stock variables.

It is not practical to apply manually the predicate logic formulae for any case modelling. Rather, it should be done by computer software because the applying process is clearly defined and contains repetition. Predicate logic formulae can be a program in Prolog language without technical changes. Once the information of variable dependencies is followed by the definition, each variable's time property can be listed.

In the case with predefinition of stock variables, a Prolog program from formulae (2) and (3), methods of defining each variable's dependencies and time properties predefinitions are shown in figure 4. Using the program in figure 4 as a part of Prolog programs which express the structure of causal loop diagrams and partly indicate time properties, stock flow diagrams can be obtained.

Dependency definition (*L(Y, X)* in logic formula)

```
variable(X, Y).
```

Time Property definition (*P(X, Y)* in logic formula)

```
property(X, Y).
```

Flow variable definition

```
property(X, flow):-
       variable(W, X), property(W, stock),
       not(property(X, auxiliary)).
```

Auxiliary variable definition

```
property(X, auxiliary):-
       variable(V, X),
       not(property(V, stock)),
       not(X=''), %for display control
       not(property(X, stock)),not(property(X, flow)).
```

Figure 4: Translation from predicate logic formulae to Prolog program

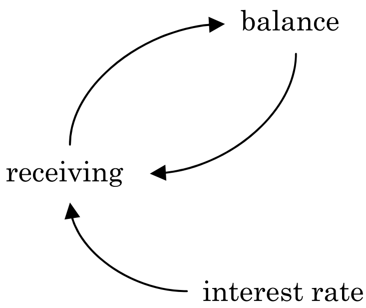


Figure 5: causal loop diagram of a bank account model

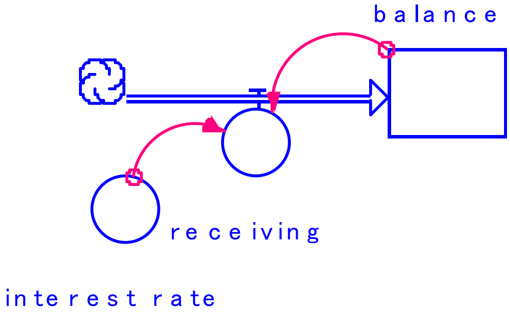


Figure 6: Stock flow diagram of a bank account model

## 3. Application

In order to examine the validity of the process of giving dynamic structure to causal loop diagrams using a Prolog program, this section applies two cases to the process.

Firstly, one of the simplest cases, shown in figure 5, is examined. This case expresses a bank account balance. An account holder deposits his/her salary into his/her account monthly. The quantity of it is indicated as "receiving." The bank will give him/her interest according to a fixed interest rate.

Now, a model builder decides that the variable "balance" is a stock variable. Therefore, a Prolog program can be implemented in the process shown in the previous section. Giving a question to the Prolog interpreter, one can obtain information about each variable's dependencies and time property. This question and answers are shown in the Appendix. From the answer given by the Prolog interpreter, model builders can make stock flow models, shown in figure 6.

Secondly, a larger case in Takahashi (2005B) is examined. This model shows that the extra time labour tendency is not a national characteristic (or industriousness) but a difference of a social support system for its members. More details are shown in Takahashi (2005B). The causal loop diagram is shown in figure 7. This diagram seems complicated so that it is not easy to convert it to stock flow diagram manually. However, it is not problematic when computer software does it. The Prolog program for it is the same except for each variable definition. The result of query for the Prolog interpreter is shown in figure 8. This diagram is the same as a diagram in the original text without the variables' directions. Causal loops do not originally have flow directions so that this automatic process cannot define the directions. Therefore, figure 8 indicates that all

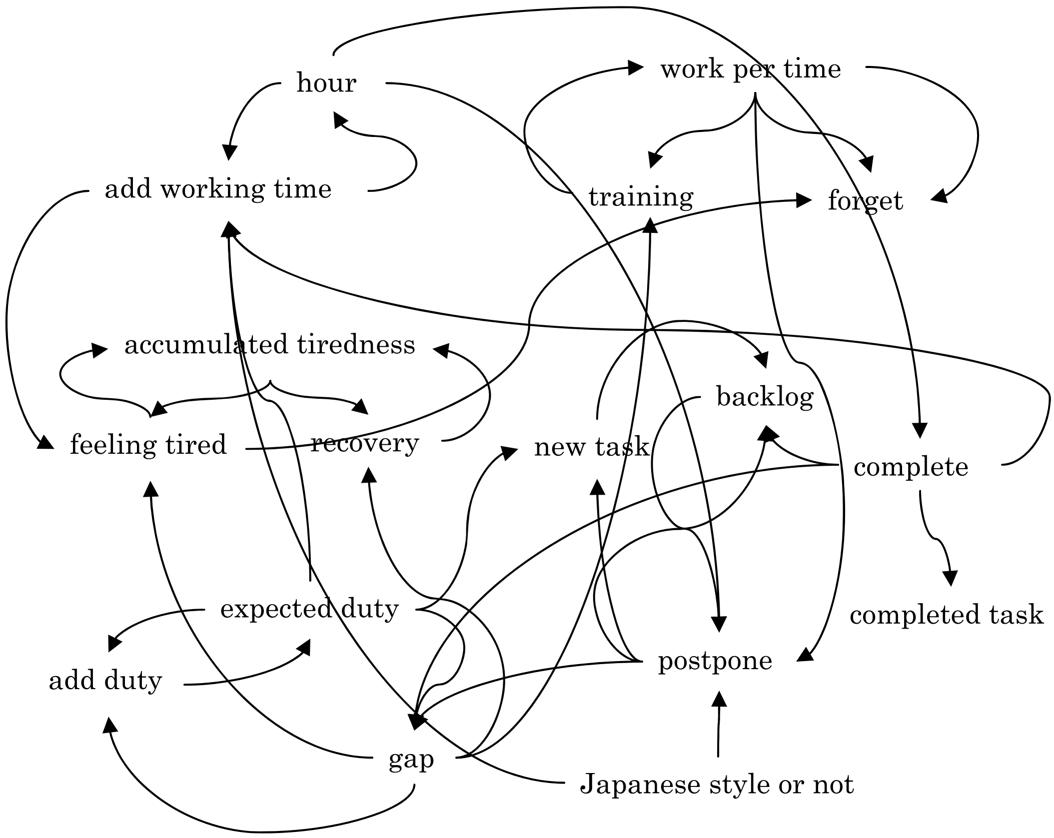


Figure 7: Causal loop diagram of Extra time work model

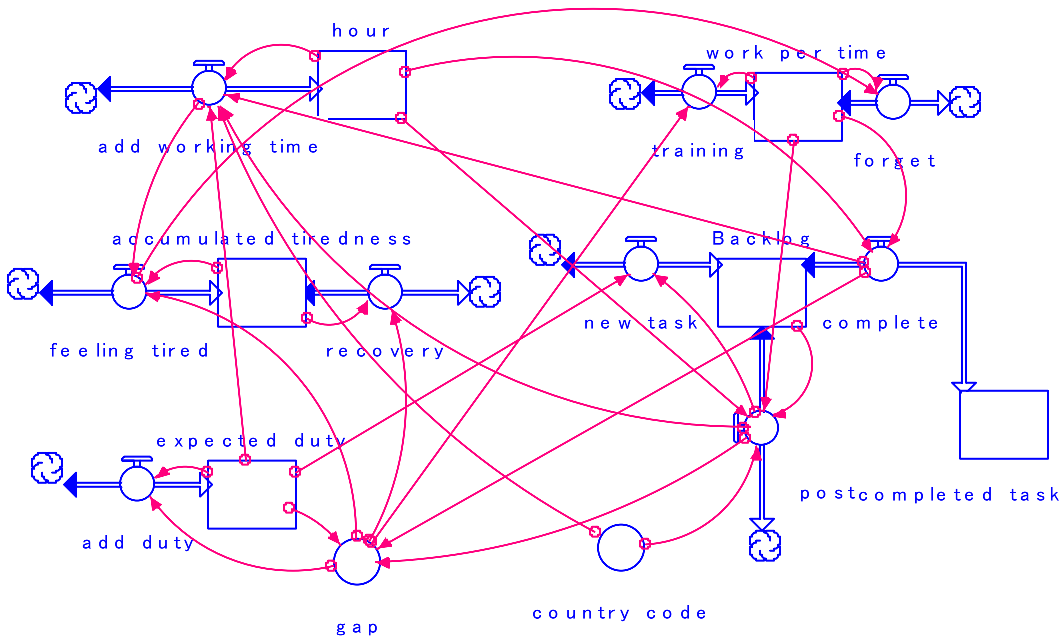


Figure 8: Stock flow diagram of Extra time work model

flow variables are bi-flow. Model builders need to define the directions by meanings of models and elements. In figure 8 case, the original model definition contains only a definition in which variables are stock variables.

Thus, System Dynamics model building is possible using limited or incomplete information.

## 4. Conclusion

This research showed a way of making automatically stock flow diagrams which are necessary to make numerical simulations from our mental models through causal loop diagrams. Causal loop diagrams do not have dynamic information. In other words, information at the level of drawing or reading causal loop diagrams is incomplete for numerical dynamic simulations. However, causal loop diagrams are easy to draw for most people. Receiving part of the information about the whole variables as a beginning point, the process this paper has suggested can automatically make stock flow diagrams. This also suggests System Dynamics simulation software’s implementation of functions to draw causal loops and to translate them to stock flow diagrams.

This paper has not focused on the design of the software’s user interface or software implementation methods so there is scope for improvement on them. Both of them have been already explored in each field so that such improvement can be done without difficulties. Moreover, the corresponding table in figure 3 based on Takahashi (2005) and the process suggested in this paper can contribute towards satisfactory achievement in automatic model building.

Acceleration of model building must be effective not only for modelling beginners but also business users. Time for making solutions is always limited. Making models manually with carefully can improve researchers’ abilities to build appropriate System Dynamics models. However, in business scenes, practical and real tasks after analyses should start as quickly as possible. This research has contributed to such demand.

## Appendix

This is a Prolog program which can run on most runtime environment of Prolog. It is examined on SWI-Prolog which is free software by The Human Computer Studies.

```
% variables: balance, receiving, interest_rate
setproperty(balance,stock).
variable(balance, receiving).
```

```
variable(receiving,balance).
variable(receiving,interest_rate).
variable(interest_rate,'').
% General rules
property(X,Y):-setproperty(X,Y).
property(X,flow):-variable(W,X),property(W,stock),not(property(X,a
uxiliary)).
property(X,auxiliary):-variable(V,X),not(property(V,stock)),not(X=
''),not(property(X,stock)),not(property(X,flow)).
```

Here is a result of the query which asks each variable's property and dependencies.

```
?- property(A,B),variable(A,C).

A = balance
B = stock
C = receiving ;

A = receiving
B = flow
C = balance ;

A = receiving
B = flow
C = interest_rate ;

A = interest_rate
B = auxiliary
C = '' ;
```

## References


Kainz, Daniel and Ossimitz, Günther. 2002. *Can Students learn Stock-Flow-Thinking?* International Conference of the System Dynamics Society, Palermo, Italy.

Ossimitz, Günther. 2002. *Stock-Flow-Thinking and Reading stock-flow-related Graphs.* International Conference of the System Dynamics Society, Palermo, Italy.

Richardson, George. 1997. *Problems in causal loop diagrams revisited.* System Dynamics ReviewVol 13, No. 3.

Richmond, B.1992. *An Introduction to Systems Thinking*. iseesystems.

Senge, Peter 1990. *The Fifth Discipline.* Doubleday.

Sterman, J. 2000. *Business Dynamics*. McGraw-Hill.

Sterman, J. 2002. All models are wrong: reflections on becoming a systems scientist. *System Dynamics Review* 18 (4): 501-532.

Sweeney, L. B. and Sterman, J. 2000. Bathtub Dynamics: Initial Results of a System Thinking Inventory. System Dynamics Review 16 (4): 249-286.

Takahashi, Yutaka. 2005. *Translation from Natural Language to Stock Flow Diagrams*. International System Dynamics Conference, 2005, Boston.

Takahashi, Yutaka. 2005B. *System Dynamics Analysis of Differences between Extra Time Work and Postponement*. 2005 Conference of System Dynamics and Management Science Sustainable Development of Asia Pacific, Shanghai.