電腦控制系統中模式混淆之偵測

Mode-Confusion Detection for Computer Controlled Systems

江坤達　　曾婉惠　　范金鳳
Kun-Da Chiang　Wan-Hui Tseng　Chin-Feng Fan
元智大學 資訊工程研究所
Computer Science and Engineering Department, Yuan-Ze University
csfanc@saturn.yzu.edu.tw

摘要

人機界面互動上的錯誤引發之意外事件佔整個電腦相關致命意外的比率極大[7]，因此如何避免操作員心理模式和系統工作模式的不對稱所產生的模式混淆(mode confusion)是電腦控制系統設計中的關鍵議題。

為降低發生模式混淆的機率，本研究在系統設計階段利用狀態圖(statechart)建構整體系統的模式，接著制訂條列式規則表(rule based table)，分別表達自動化系統應有的動作，以及操作員依據常識判斷而對系統做出的反應動作。藉由兩者的比對找出操作員對系統不熟悉、易出錯的地方，進而改善人機界面設計上的缺失。除此之外，在執行階段我們加入模式混淆偵測模組(mode confusion detection module)收集操作員的操作動作，用以推論操作員認知的系統模式並比對目前系統的工作模式，以偵測可能的混淆狀況，才可即時提供操作員重要的警訊，以避免因模式混淆可能造成的重大意外。

關鍵詞：模式混淆偵測、狀態圖、電腦控制系統。

Abstract

According to McKenzie [7], 92% of computer-related accidental deaths were due to human–computer interaction errors. How to prevent mode confusion caused by a mismatch between the operator's mental mode and the system's working mode is a critical issue.

This research designed methods to detect potential mode confusion. To reduce the probability of the mode confusion, statechart diagrams were used to construct the system mode in the design stage. Next, rule-based tables were established. The tables expressed the automatic system's action and the operator's control by common sense. Though the comparison, the unfamiliar and error-prone operations of the operator can be identified. Defects of the human-computer interface design can then be improved.

Besides, at run-time we added a mode-confusion detection module to first record the operator's commands, and then infer the operator's current working model to compare it with the current system working mode. A real-time alert will be issued if there exists a mismatch. Thus, accidents induced by mode confusion can be prevented.

Keyword: mode confusion detection, statechart, computer control system.

1. 簡介

電腦已經成為現今人們生活中不可缺少的一部份。其中與大眾的生命安全與財產相關的電腦系統，例如交通、核能、

太空與醫療等系統，通稱為安全關鍵性計算系統 (safety critical computing systems)。安全關鍵性軟體與一般商用軟體最大的不同在於安全關鍵性軟體與大眾的生命安全、財產有密切的關係，當軟體發生意外時，將造成的嚴重後果。通常此類型的系統多數為複合式系統(hybrid system)包含硬體、自動化電腦控制系統與操作人員三部份。

現行的自動化系統控制改變了傳統人為控制的角色。人機界面的不良設計可能引發操作員對硬體系統及自動化系統執行狀態的誤解，此種誤解稱作模式混淆(mode confusion)[4][5][6][9]。模式混淆是造成航空事故的主因之一。例如飛機處於自動航行模式時，機長卻認為飛機處於手動模式，如此的模式混淆極有可能造成飛機失控而導致意外。因此若能偵測及消除潛在的模式混淆，將可大幅減少不必要的人為損害，提高安全關鍵性軟體的可靠度及保障大眾的生命安全。故本研究發展了一套複合式系統人機介面模式混淆偵測的方法。我們的方法不僅能夠在早期系統開發階段，偵測出可能造成模式混淆的設計，亦能夠在執行階段(run time)即時偵測到模式混淆的發生。

在設計階段，我們以狀態圖和條列式規則表的方式，分別表達自動化系統應有的動作以及操作員依據常識判斷而對系統做出的反應動作，藉由兩者的比對找出操作員對系統不熟悉、易出錯的操作方式或是系統訊息可能表達不清楚的地方，我們可根據得到的結果進一步加強、改善系統人機界面的設計，以達到降低模式混淆發生的機率。

在執行階段，我們在系統中加入模式混淆偵測模組 (mode confusion detection module)，藉由擷取操作員的反應動作與模組內儲存的模式混淆偵測模組資料庫，來預測操作員內心認為的系統模式，並比較是否與系統目前的模式相左，可即時給予操作員相關的警訊，進而避免因模式混淆所造成的重大意外事件。

第二章為相關的背景研究，第三章說明我們的方法論，在第四章我們以鍋爐系統作為個案的研究與分析，最後提出簡單的結論。

2. 研究背景

2.1 模式混淆(mode confusion)

當操作員對系統現行的模式不清楚時，稱為模式混淆(mode confusion)。操作員常因知識不足或對現狀錯誤的認知引發系統的意外。Leveson[3][4][5][8]將安全關鍵系統中設計缺失所造成的潛在模式混淆，分為六類，分述如下：

- 界面解讀錯誤(Interface Interpretation Errors)：系統對使用者輸入的值產生誤解並給予錯誤的輸出值，因而造成錯誤。
- 不一致行為(Inconsistent Behavior)：當系統中存在不一致的行為，使得相同的輸入卻造成不同的行為結果。
- 間接模式改變 (Indirect Mode Changes)：系統沒有收到操作員明確的指令卻自動轉換模式時，則稱為間接模式改變，通常這樣的模式轉換會造成意外事故的發生。
- 非預期的副作用 (Unintended Side Effects)：某些操作行為可能引發操作員未知的額外效應（副作用），意外因而產生。
- 缺少合適回饋(Lack of Appropriate Feedback)：系統回饋的資訊不完整而造成操作員對操作模式混淆，進而引發錯誤。
- 操作員權限限制(Operator Authority Limits)：內鎖(interlock)等系統保護動

作，讓操作員在緊急狀況下，無法得到適當權限。

現行模式偵測(mode detection)的方法不多，更是未見能自動化的偵測方法。本研究發展一套有系統的事前和執行階段的模式混淆偵測方法，以改善人機介面設計的安全性。

3. 研究方法

我們的方法包含事前設計階段的分析及執行時的即時偵測兩大部分，詳述如下。

3.1 設計階段的模式混淆偵測方法

首先我們發展潛在模式混淆偵測的分析方法。第一步驟我們將系統以視覺化的方式來建構，在此我們利用狀態圖(statechart)建構整體系統的模式。接著根據狀態圖中的各種模式，我們依據模式內不同的感應組合值制訂條列式規則表(rule-based table)，分別表達自動化系統應有的動作以及操作員依據常識判斷而對系統做出的反應動作。藉由兩者的比對找出人機界面上設計的缺失，以便加強人機界面的互動性。

由於狀態圖具有階層式(hierarchical)的表達特性，所以我們利用其特性將系統分成兩層不同角度的狀態圖，第一層為主要模式狀態圖(Primary mode state diagram)，第二層為感應值組合狀態圖(Combination of sensor state diagrams)。

第一層的主要模式狀態圖包含各種正常和異常的狀態模式，表達整體系統的模式。第二層的感應值組合狀態圖表達系統的細部狀態。我們將系統每一種感應器的(sensor)量測數值劃分成數個區段，例如將量測水位區分成高中低三種水位，並依照不同感應器的量測區段來進行組合，這樣的組合表示系統目前的狀況。每一種感應組合值只對應到一種系統模式，而系統模式則允許有多種的感應組合值，例如「高水位、高壓力、高溫度」就是一組感應組合值，將各種組合歸類到系統模式中。其中不同的感應值組合狀態會有其對應的反應動作。

當建構好系統模式架構圖後，我們以條列式規則表(rule-based table)依據不同的狀態和感應組合值來表示系統的反應動作以及操作員的反應動作。在規則表的定義中，表頭的資料表示系統的主要模式（狀態）和感應值組合，接著列出系統反應、硬體設備設定狀態及所對應的操作員或自動化系統的反應動作，如表 1 所示。

表 1 條列式規則表

系統模式(State)		感應組合值		
1	系統反應	壓力緩緩升降	溫度快速下降	水位快速上升
硬體設備設定狀態		給水閥：功率大、加熱器：功率小、蒸氣閥：on		
系統反應動作 或 操作員反應動作		將給水閥功率降一級		

當自動化系統反應動作和操作員反應動作以表格的方式呈現後，第一，可採用人工的方式來比對相同的狀態條件下是否擁有相同的操作動作，以找出潛在的模式混淆，第二，若是狀態條件很多，則可以將系統模式、量測組合和其他狀況以代碼表示，並儲存於資料庫中，以便將來用於電腦搜尋、排列和比對。 其中自動化系統反應動作可從軟體規格和需求得到。而操作員反應動作主要可根據操作員對於系統的常識(common sense)作為判斷依據。

3.2 執行階段的模式混淆偵測方法

在人機介面模式混淆的議題中，除了事前找出可能發生的問題以外，如何在執行階段也能夠預測或是偵測可能的問題也是一項相當重要的課題。所以我們在系

統中加入模式混淆偵測模組(mode confusion detection module)來幫助系統和操作員之間的運作，圖 1 為我們所提出的系統架構圖。

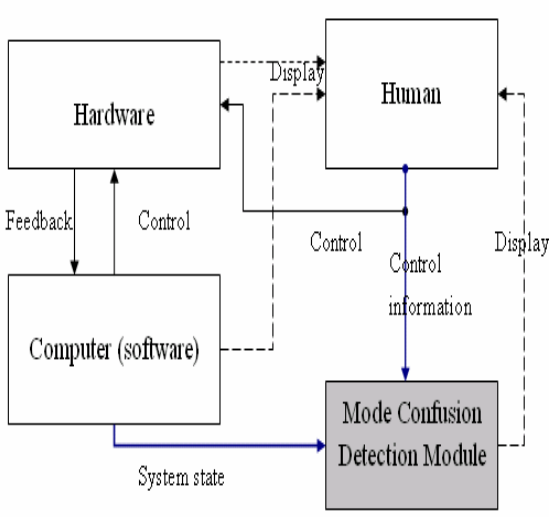

圖 1. 系統架構圖

圖 1 中系統的硬體可以藉由電腦自動化或操作員來控制，由於系統動作的執行會使得系統狀態改變並回饋給電腦，電腦再將訊息顯示給操作員，根據不同的訊息，操作員便可以判斷下一操作步驟。在圖中因此我們將模式混淆偵測模組加入在電腦和操作員之間，一方面收集系統目前的狀況，另一方面擷取操作員控制系統的指令，其中在模組內部的比對方式是藉由擷取操作員的指令動作，來推論操作員認為系統目前的模式，並加以比對系統目前的狀況和模式，若是推論的模式和系統目前的模式並不一致，則可能是產生了模式混淆，此時必須即時給予操作員適當警訊，避免模式混淆的情況發生。

由操作員動作推論操作員的認知，可能產生多種推論結果。舉例來說，操作員可以在正常水位時，希望水位上升，也可以在低水位時，希望水位上升，因此水位上升是我們希望系統完成的動作，而正常水位和低水位卻是系統兩種不同的狀態模式，所以我們必須進一步分析系統的狀況條件，來判斷出系統的改變是在哪一狀態模式中。故我們希望模式混淆偵測模組能夠由操作員動作來推論操作員的內心模式。我們的步驟如下：

1. 比對已建立的模式混淆偵測模組資料庫，一旦發現模式混淆，則以訊息回應操作員，否則繼續步驟 2~4。
2. 把所有的硬體動作組合全部列出，假設系統中有四個硬體設備，每種硬體設備有三種不同的操作，則系統共會有 81 種的動作組合。
3. 每種動作組合必定對應一種目的。例如打開給水閥是為了使鍋爐水位上升，而這樣的目的必定對應一個以上的系統狀態模式。舉例而言，在正常模式可以加水，在低水位模式也可以加水，所以，不同的動作組合將可對應到系統不同的模式，因此我們可以條列出動作組合與對應的系統模式。
4. 最後分析感應器得到的訊息值，便可以推論出操作員所認為的模式，若是推論出來的模式與系統的現行模式有所差異，則產生警示。

4. 實作案例

本章中我們將以鍋爐(steam boiler)的案例來說明我們的方法。

4.1 案例:蒸氣鍋爐系統

4.1.1 系統描述

針對鍋爐系統硬體和環境(Physical environment)描述[1][2]：

A. 鍋爐(steam-boiler)：鍋爐內定義四個水位，包含最高水位(M2)、最低水位(M1)、最大正常水位(N2)與最小正常水位(N1)限制，當鍋爐正常運作時，水位應處於最大正常與最小正常水位之間，若是水位高於最高水位或是低於

最低水位時，將會造成意外事故發生。

B. 給水閥(pump)：提供鍋爐進水控制。

C. 洩水閥(valve)：提供鍋爐出水控制。

D. 加熱器(heater)：提供鍋爐爐水加熱。

E. 空氣閥：調節鍋爐內壓力。

F. 蒸氣閥：蒸氣出口。

G. 感應器(sensor)：

- 水位感應器(water-level sensor, W)：量測鍋爐內水位。
- 溫度感應器(temperature sensor, T)：量測鍋爐水溫度。
- 壓力感應器(pressure sensor, P)：量測鍋爐內壓力，鍋爐內可承受的最大壓力為P1，最大正常承受壓力為P2。

4.1.2 設計階段的模式混淆偵測

第一步驟先建構系統主要模式狀態圖(Primary mode state diagram)，其中鍋爐系統包含七個主要模式(mode)，如圖2所示，分別為：

- 測試模式(test mode)：測試鍋爐內的硬體是否正常。
- 啟動鍋爐模式(start boiler mode)：將鍋爐注水到一定量，並加熱直到產生蒸氣。
- 正常模式(normal mode)：水位和壓力皆處於正常範圍的標準操作模式。
- 減水模式(reduce mode)：當水位低於最小正常水位(N1)時，則必須加水。
- 提升模式(enhance mode)：當水位高於最大正常水位(N2)時，則必須釋放鍋爐內的水。
- 超壓模式(over-pressure mode)：當壓力超過設定的最大正常承受壓力(P1)時，則必須釋放壓力。
- 危急模式(emergency mode)：　當水位超過最高水位、低於最低水位或是壓力高於最大承受壓力時，則進入危急模式。

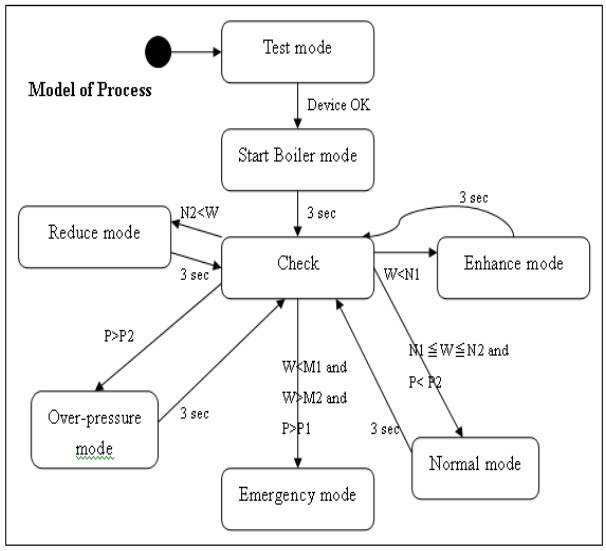

圖 2：主要模式狀態圖

根據系統中的感應器建立第二層的感應值組合狀態圖(Combination of sensor state diagrams)，依照系統三種不同感應器所量測出來的不同條件狀態來組合鍋爐的狀況，在此我們以系統模式為正常模式下的“壓力正常、溫度>100度與正常水位”為探討的案例。

我們對系統模式為正常模式，而感應組合為“壓力正常、溫度>100度與正常水位”的狀態下，由系統的需求規格分析出對應的條列式規則表(rule-based table)來表達自動化系統的操作方式，如表2所示。例如當系統的反應是“壓力逐漸上升、溫度快速下降、水位快速上升”且硬體的狀態為“給水閥：功率大、加熱器：功率小、蒸氣閥：on、洩水閥：off、空氣閥：off”時，此時自動化系統的反應動作則為“給水閥功率降一級”。接下來，我們利用操作員的常識來分析操作員的模式，如表3為操作員對應的條列式規則表。

表 2 自動化系統條列式規則表

系統模式(Normal)	壓力正常	溫度> 100 度	正常水位
1 系統反應	壓力緩緩升降	溫度快速下降	水位快速上升
硬體設備設定狀態	給水閥：功率大、加熱器：功率小、蒸氣閥：on、洩水閥：off、空氣閥：off		
系統反應動作	將給水閥功率降一級		
2 系統反應	壓力緩緩升降	溫度緩緩升降	水位快速上升
硬體設備設定狀態	給水閥：功率大、加熱器：功率中、蒸氣閥：on、洩水閥：off、空氣閥：off		
系統反應動作	將給水閥功率降一級且 3 秒後系統再檢查		
3 系統反應	壓力緩緩升降	溫度緩緩升降	水位緩緩升降
硬體設備設定狀態	給水閥：功率中、加熱器：功率中、蒸氣閥：on、洩水閥：off、空氣閥：off		
系統反應動作	不動作且 3 秒後系統再檢查		
4 系統反應	壓力緩緩升降	溫度緩緩升降	水位快速下降
硬體設備設定狀態	給水閥：功率小、加熱器：功率中、蒸氣閥、on、洩水閥：off、空氣閥：off		
系統反應動作	將給水閥功率升一級且 3 秒後系統再檢查		
5 系統反應	壓力緩緩上升	溫度快速上升	水位快速下降
硬體設備設定狀態	給水閥：功率小、加熱器：功率大、蒸氣閥：on、洩水閥：off、空氣閥：off		
系統反應動作	將給水閥功率升一級且 3 秒後系統再檢查		
6 系統反應	壓力緩緩上升	溫度快速上升	水位緩緩升降
硬體設備設定狀態	給水閥：功率中、加熱器：功率大、蒸氣閥：on、洩水閥：off、空氣閥：off		
系統反應動作	將加熱器功率降一級且 3 秒後系統再檢查		

表 3 操作員條列式規則表

系統狀態(Normal)	壓力正常	溫度> 100 度	正常水位
1 系統反應	壓力緩緩升降	溫度快速下降	水位快速上升
硬體設備設定狀態	給水閥：功率大、加熱器：功率小、蒸氣閥：on、洩水閥：off、空氣閥：off		
操作員反應動作	將給水閥功率降一級		
2 系統反應	壓力緩緩升降	溫度緩緩升降	水位快速上升
硬體設備設定狀態	給水閥：功率大、加熱器：功率中、蒸氣閥：on、洩水閥：off、空氣閥：off		
操作員反應動作	將給水閥功率降一級		
3 系統反應	壓力緩緩升降	溫度緩緩升降	水位緩緩升降
硬體設備設定狀態	給水閥：功率中、加熱器：功率中、蒸氣閥：on、洩水閥：off、空氣閥：off		
操作員反應動作	不動作		
4 系統反應	壓力緩緩升降	溫度緩緩升降	水位快速下降
硬體設備設定狀態	給水閥：功率小、加熱器：功率中、蒸氣閥、on、洩水閥：off、空氣閥：off		
操作員反應動作	將給水閥功率升一級		
5 系統反應	壓力緩緩上升	溫度快速上升	水位快速下降
硬體設備設定狀態	給水閥：功率小、加熱器：功率大、蒸氣閥：on、洩水閥：off、空氣閥：off		
操作員反應動作	將給水閥功率升一級		
6 系統反應	壓力緩緩上升	溫度快速上升	水位緩緩升降
硬體設備設定狀態	給水閥：功率中、加熱器：功率大、蒸氣閥：on、洩水閥：off、空氣閥：off		
操作員反應動作	不動作		

比較自動化系統的條列式規則表(表 2)與操作員的條列式規則表(表 3)，我們可以得知自動化系統反應與操作員反應有所出入，其中第一個混淆地帶為操作員在每一種操作動作後，必須要有 3 秒鐘的等待回應時間，但是表 3 的操作員反應動作中完全疏忽了。第二個是系統在正常模式中，溫度和水位同時有問題時，則必須以溫度為優先來考慮，在表 3 編號 5 中，卻只考慮到水位。另外第三個是為了防止快速熱漲冷縮對鍋爐產生的不良影響，所以當溫度快速上升時，操作員必須有相對應的動作，但是表 3 中的編號 6，操作員並沒有任何的反應動作；因此藉由條列式規則表(rule-based table)的方式，在設計階段便可發現模式混淆的地方，如此一來便可進一步地改善設計上的缺失，加強人機界面設計的明確性。

4.1.4 執行階段的模式混淆偵測

在執行階段的偵測，首先我們在系統中加入模式混淆偵測模組(mode confusion detection module)。藉由擷取操作員的指令動作，來推論操作員認為系統目前的模式，在模組中我們以硬體動作對應系統模式資料庫為主要的推論依據，如表 4 所示。

表 4 模式混淆偵測模組資料庫

洩水閥	給水閥	加熱器	推論模式(inferred mode)
關(off)	功率大	功率大	Normal
		功率小	Enhance **or** Over-pressure
		關(off)	Initialization
	功率中	功率大	Normal
		功率小	Enhance **or** Over-pressure
		關(off)	Initialization
	功率小	功率大	Normal
		功率小	Enhance **or** Over-pressure
		關(off)	Initialization
	關(off)	功率大	Normal
		功率小	Over-pressure
		關(off)	Emergency
開(on)	關(off)	功率大	Reduce
		功率小	Reduce
		關(off)	Emergency
開(on)	關(off)	關(off)	Error operation

每一種硬體設備的動作都會對應到一種到兩種模式。對應兩種模式的動作例子如操作員的動作為“洩水閥為關、給水閥轉至大且加熱器功率為小”時，則在模組中可能對應的是提升模式(Enhance mode)和超壓模式(Over-pressure mode)，接下來此模組便根據判斷感應器所測量的數值來推論，若是壓力處於正常狀況下，則推論為提升模式，反之則推論為超壓模式。

鍋爐系統的模擬，如圖 3 所示。我們利用此模擬系統[1]，模擬模式混淆的偵測。模式混淆可能因操作員的疏忽，也可能是硬體的問題，更可能是操作員的訓練不足。因此此模擬系統不僅能夠即時偵測模式的混淆，也能夠訓練操作員正確的操作反應，提高操作員對於系統內部運作的認知。

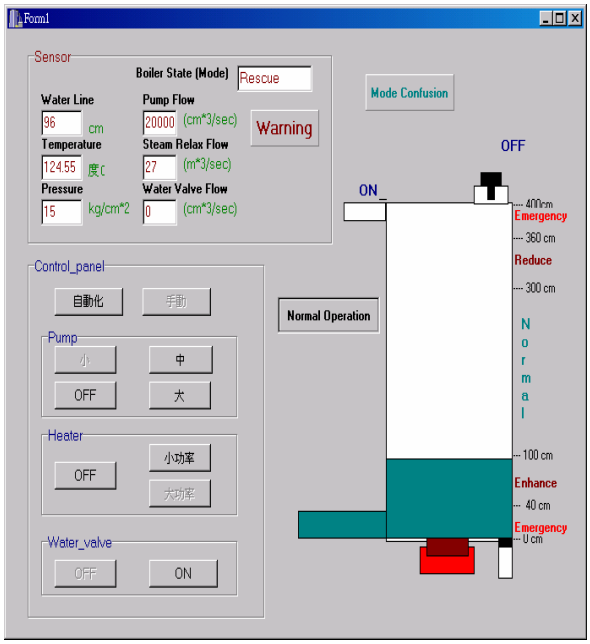

圖 3 鍋爐系統模擬

5. 結論

複合式系統的人機介面可能隱藏潛在模式混淆的地方，除了操作員誤解資訊的涵意外，缺乏即時的回應訊息一樣會導致人為操作錯誤。本研究提出一套複合式系統人機介面模式混淆偵測的方法，利用狀態圖與條列式規則表的方式，不僅能夠在早期系統開發階段，偵測系統中潛在的模式混淆設計，加強並改善系統人機界面設計上的缺失；亦能夠在執行階段即時偵測模式混淆的發生，盡早通知操作人員相關的警訊，以降低模式混淆發生的機率，避免因模式混淆所造成的重大危害事件。

6. 參考文獻

[1] 江坤達, “電腦控制系統中模式混淆偵測之發展與應用”,元智大學資訊工程研究所,碩士論文, 2004, 六月。

[2] Jean- Raymond Abrial, “Steam-boiler control specification problem,” August 10, 1994.

[3] E. Bachelder et al. “Describing and Probing Complex System Behavior: A Graphical Approach,” In the proceedings of the Aviation Safety Conference, Seattle, Sept. 2001.

[4] R.W. Butler, S.P. Miller, J. N. Potts, V.A. Carreno, “A Formal Methods Approach to the Analysis of Mode Confusion,” Proc. of 17th Digital Avionics Systems conference, Vol 1,pp.C41/1-C41/8, 1998.

[5] N. G. Leveson et al. “Analyzing Software Specifications for Mode Confusion Potential,” Proceedings of the Workshop on Human Error and System Development, Glascow, March 1997.

[6] N. G. Leveson and E. Palmer. “Designing Automation to Reduce Operator Errors,” In the Proceedings of Systems, Man, and Cybernetics Conference, Oct. 1997

[7] D. Mackenzie “Computer-related accidental death: an empirical exploration,” *Science and Public Policy*, vol 21 No(4), pp.233-248, 1994.

[8] J. Rushby, J. Crow, E. Palmer, "An Automated Method to Detect Potential Mode Confusions," Proc. of 17th Digital Avionics Systems conference, vol 1/17, pp. 4.B.2-1-4.B.2-6, 1999.

[9] Mario Rodriguez et al. "Identifying Mode Confusion Potential in Software Design," Digital Aviation Systems Conference, October 2000.