Analysis of Stakeholder Concerns for Vehicle Navigation System Architecture Solution

Xiaoying Kong
Faculty of Engineering and Information Technology
University of Technology, Sydney

Li Liu
Faculty of Engineering and Information Technologies
The University of Sydney
Sydney, Australia

Tich Phuoc Tran
Computer Science and Information System
Victoria University, Sydney.
Sydney, Australia

Kumbesan Sandrasegaran
Faculty of Engineering and Information Technology
University of Technology, Sydney, Australia

***Abstract*— This research presents a plan-driven development methodology in early stage of vehicle navigation system development. Different from conventional flow, an architecture prerogatives phase is designed before vehicle navigation system specifications. In this approach, stakeholder concerns are identified and analyzed. A solution architecture is derived from the stakeholder concerns. The considered architectures are selected from best practice GNSS/INS integrated system architecture patterns. The sensor integration architecture decision is provided to all developers as a vision before the start of detailed specifications. This makes the final system specification more supportable by all stakeholders.**

Keywords*— development methodology, GNSS/INS integration, sensor fusion pattern, vehicle navigation design

I. INTRODUCTION

In literature most of the research efforts on vehicle navigation systems have been focusing on navigation sensor technologies and sensor fusion algorithms. The research outcomes of these technologies will be applied in vehicle navigation system development for users.

In navigation sensor technologies, two types of sensors have been developed and applied: absolute sensors and dead reckoning sensors [1]. Among absolute sensors, Global Navigation Satellite System (GNSS) is widely used in recently years. Inertial Measurement Unit (IMU) is largely used as dead reckoning sensors. Major efforts on these sensors are sensor hardware development and quality improvement [2]. At navigation system level, sensor fusion algorithms have been studied and applied for real time navigation system operations. These sensor fusion algorithms are the core in the navigation system architecture. In research side, sensor technologies and sensor fusion algorithms are built into prototypes to verify the research results. In industry development of vehicle navigation systems, some of the prototypes and research results are chosen and applied.

To effectively apply these technologies to industrial vehicle navigation system development, there is a need to develop methodologies from system engineering view point. Vehicle navigation systems are both hardware and software-intensive systems. To avoid project failures in software industry, software engineering processes should be considered. There are two types of major software development methodologies: heavy weight plan-driven processes, and light-weight agile methodologies [3]. In plan-driven processes, the basic model is waterfall process. Waterfall activity flow includes: project planning, requirements and specifications, design, implementation, verification and validation, system operation and maintenance. Plan-driven processes try to use activities and associate artifacts to constraint project and guaranty the final product quality. Agile methodologies apply informal methods, incrementally deliver part of system, minimize documentations, response to changes.

Vehicle navigation systems are life-critical systems. Safety requirements have the highest priority in system specifications. We would not recommend agile methodologies to be used in the development process of vehicle system. In literature of vehicle navigation development methodologies, Kong et al. presented an entity relationship model to analyze the vehicle navigation research literature [2]. A process model including the development flow and associated artifacts in sensor selection decision making is demonstrated. Guo et al. [4] proposed a requirement-design framework of vehicle navigation integration. This framework models artifacts and their traceable relationships in requirement and design stages.

To fill in the gap of vehicle navigation system development process flow, in this research we develop a process to guide what activities and what artifacts development team could produce in early stage of development. A plan-driven "*Architectural Prerogatives*" process model is designed. This process uses a flow of architectural prerogatives/specifications/design to replace the requirements/specification/design stages of waterfall model. The aim of this process model is to provide a vision of system architecture before system specification. This approach makes the final navigation system more supportable by stakeholders.

This architectural prerogatives model is presented in this paper as follows. Section II presents the process model of architectural prerogatives. Section III details the phases of stakeholder analysis for vehicle navigation systems. Section IV demonstrates the navigation system architecture evaluation and decision approach. Conclusion and future study are drawn in the last section.

II. ARCHITECTURAL PREROGATIVES MODEL

The basic plan-driven process starts from planning, specifications, design and down to implementation and system testing. As argued by Maclaszek, before start the detailed system specification, development team should adopt architectural patterns and principles understandable to all developers. Without a vision of system architecture, delivered system specifications will be unsupported [5]. We adopt this theory and add an "Architectural Prerogatives" phase before the phase of navigation system specifications. The development flow in early stages can be presented using the flow in Fig. 1.

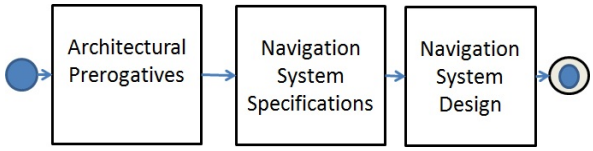

Fig. 1. Process flow: from navigation system architectural prerogatives to specifications and design

Inside the architectural prerogatives, we design a set of sub phases: identify stakeholders of navigation system; elicit stakeholder concerns; analyze quality attributes/ development constraints; navigation system architecture consideration; navigation architecture evaluation and decision. Fig. 2 illustrated the flows of the architectural prerogatives. In the following sections, the details of each sub phase of this process model will be presented.

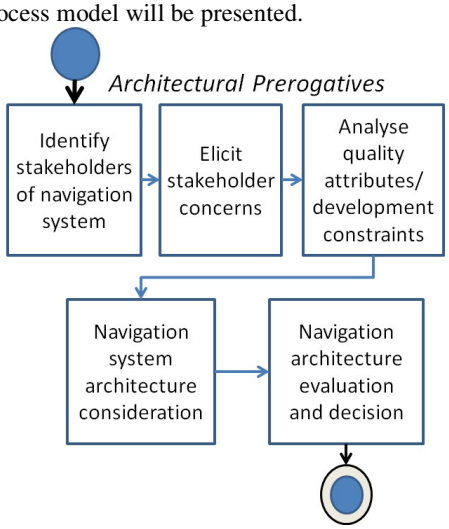

Fig. 2. Process for stakeholder analysis and architecture evaluation

III. STAKEHOLDER ANALYSIS

A. *Process flow for analysis of stakeholder concern and architecture evaluation*

Navigation products are built to meet user requirements. User requirements include functional requirements and non-functional requirements. Non-functional requirements are system quality attributes and development constraints that can be derived from stakeholder concerns. These quality attributes and constraints are analyzed and built in to navigation system architecture. System functional requirements can be transferred into modules in architecture.

During the stage of identifying stakeholders, people who will involve in the navigation project are listed. Usually the stakeholders for a navigation system include: business owner of the navigation system, project manager for coordinating the system development, system engineers to research and develop the system, system end users who operate the system, and system maintainers to make changes to the system. These stakeholders stand in their own view points and have different concerns. Their concerns are elicited and will be classified into quality attributes and development constraints. These quality attributes and constraints then become the basis for design, evaluation and selection of navigation system architectures.

In this section, we will present the approach of elicitation of stakeholder concerns, and analysis of quality attributes and development concerns. In next section, navigation system architecture consideration and evaluation will be presented.

B. *Elicitation of stakeholder concerns*

Stakeholder concerns can be identified by research and survey on existing system documentations and interview with stakeholders. To build navigation systems for market users, stakeholder concerns can be elicited in interview. Here we use an example of a navigation system to demonstrate the sample concerns from stakeholders using elicitation process and review literatures [1, 4, 6-11]. The sample system is an aircraft navigation system using GPS and INS integration.

Sample concerns of business owner are: low cost of system development, high quality of final navigation system to satisfy market users, and on time delivery of final product.

Project managers may concern: on time delivery of final product; simple design and implementation; easy system maintenance; high accuracy, high reliability and robust navigation system; easy product use for end users; and easy collaboration with suppliers.

System engineers concern: high signal processing performance of hardware; selection of high quality GPS receiver and Inertial Measurement Unit; effective system structure configuration for GPS and IMU integration; effective efforts in designing algorithms to integrate GPS and IMU; on time delivery of final system; selection of optimal integration architecture to achieve high accuracy, reliable and robust navigation system; easy to repair system hardware, easy to change system modules, algorithms, data structure, and associated software coding for future maintenance.

Sample system end users concerns are easy to learn user interface of the navigation system, logical procedure of navigation command, effective fault detection, recovery, minimal vehicle path error, and continuous navigation signals.

System maintainers' concerns may include: less effort to analyze causes of failure, change system software, repair hardware, and validate changed parts to meet system requirements.

These stakeholder concerns are analyzed using quality attributes and development constraints. These are equivalent to system non-functional requirements. In literature, Reekie et al. identified and classified system non-functional requirements of a general system into runtime quality attributes and non-runtime quality attributes [12]. Runtime attributes include: performance, usability, reliability and security. Non-runtime quality attributes include: maintainability, testability, reusability, configurability and scalability [12]. Guo et al. develop a full list of quality attributes and development constraints for vehicle navigation systems [4].

In our approach, for a vehicle navigation system, we customize and classify these non-functional requirements into quality attributes and constraints as in Table I.

TABLE I. QUALITY ATTRIBUTES AND DEVELOPMENT CONSTRAINTS OF VEHICLE NAVIGATION SYSTEMS

Classification	Attributes
Quality attributes	• Accuracy • Availability • Reliability (fault tolerance / recoverability) • Robust • Safety • Security • Response time
Development constraints	• Maintainability (analyzability; changeability; testability) • Usability (learnability; user-friendly; operability) • Development complexity • Cost constraint (hardware, development cost) • Time constraint • Client and supplier collaboration capacity

To analyze the quality attributes and development constraints from stakeholder concerns, we design a traceability mechanism. Table II provides an analysis tool using traceability matrix.

TABLE II. TRACEABILITY MATRIX: ANALYSIS OF STAKEHOLDER CONCERNS

Stakeholder	Stakeholder concerns	quality attributes and development constraints
Business owner	low cost of system development	• Cost constraint • Development productivity
	on time delivery of final product	• Time constraint • Development productivity
	high quality of final navigation system to satisfy market users	• Accuracy • Reliability • Robustness • Response time
Project manager	on time delivery of final product	• Time constraint • Development productivity
	simple design and implementation	• Maintainability • Cost constraint • Time constraint • Development productivity • Client collaboration capacity
	easy system maintenance	• Maintainability • Cost constraint
	high accuracy, high reliability and robust navigation system	• Accuracy • Reliability • Robust • Response time
	easy product use for end users	• Usability
	easy collaboration with suppliers	• Supplier collaboration capacity
System engineer	high signal processing performance of hardware	• Response time
	selection of high quality GPS receiver and Inertial Measurement Unit	• Accuracy • Reliability • Robustness • Response time • Development cost
	effective system structure configuration for GPS and IMU integration	• Development cost • Maintainability
	effective efforts in designing algorithms to integrate GPS and IMU	• Development cost • Development complexity • Time constraint
	on time delivery of final system	• Time constraint
	selection of optimal integration architecture to achieve high accuracy, reliable and robust navigation system	• Accuracy • Reliability • Robustness • Response time • Development complexity
	easy to repair system hardware, easy to change system modules, algorithms, data structure, and associated software coding for future maintenance	• Maintainability • Development complexity • Client collaboration capacity
System end user	easy to learn user interface of the navigation system	• Usability– learnability
	logical procedure of navigation command	• Usability - user-friendly • Usability - learnability
	effective fault detection and recovery	• Usability – operability • Reliability – fault tolerance • Reliability - recoverability • Robustness
	minimal vehicle path error	• Accuracy • Reliability
	continuous navigation signals	• Reliability • Response time

Stakeholder	Stakeholder concerns	quality attributes and development constraints
System maintainer	less effort to analyze causes of failure, change system software, repair hardware, and validate changed parts to meet system requirements.	• Maintainability – analyzability • Maintainability – changeability • Maintainability - testability

In this tool each stakeholder concerns are analyzed using the quality attributes and constraints in Table I. These attributes will be used in architecture solution.

IV. NAVIGATION SYSTEM ARCHITECTURE EVALUATION AND DECISION

Once the stakeholder concerns are analyzed, they should be built into navigation system architecture. Architecture is defined by IEEE standard as "The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [13]. The major components of a vehicle navigation system using GNSS and INS include GNSS/INS integration component, user interface component (UI), and security component. In this research, we focus on the architecting process for core GNSS/INS integration architecture. UI and security components and their relationships to each other will be studied in further research.

From best practice of research and development in sensor integration, a number of integration architecture patterns have been formed. In our approach, during stage of architecture prerogatives, we select and evaluate architecture patterns. The characteristics, quality attributes and development constraints of selected architectures are analyzed. Stakeholders evaluate the architectures based on their concerns from different viewpoints. Architecture decision is made based on the stakeholders' evaluation. This evaluation approach is demonstrated in this section.

A. *Navigation architecture consideration*

GNSS/INS integration architecture patterns include uncoupled integration, loosely coupled integration, tightly coupled configuration, and ultra-tight integration [1,2,4,14,15]. To demonstrate our approach of how to analyze integration architecture patterns, we choose two types of architectures: *direct feedforward architecture of loosely coupled pattern*, and *tightly coupled integration architecture*. Other patterns will be analyzed in further study [1,2,4,14].

Loosely coupled architecture has the advantages of highly modular in accuracy and cost [1]. There are variant types of loosely coupled configurations: direct feedforward, direct feedback, indirect feedback, and indirect feedforward structures.

Direct feedforward architecture integrates raw inertial sensor data and GNSS observations into a non-linear filter configuration. The filter estimates vehicle position, velocity and attitude directly and feedforward this information set to system end users. See Fig. 3.

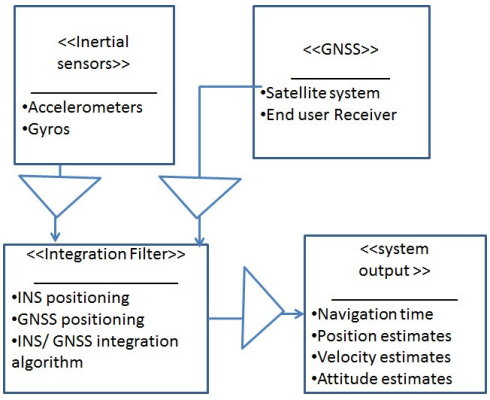

Fig. 3. Direct feedforward architecture

Direct architecture estimates and outputs navigation information at inertial sensor's sampling time. This requires high rate of filter process, and high process performance of hardware [1]. This would result more investment in hardware. High frequency of filter output would increase information availability to system end users. Inertial sensor errors are input into integration filter. Under this feedforward architecture, large unbound positioning errors will grow over time. This results in low positioning accuracy in long term [1,2,15]. Inertial navigation and GNSS navigation are independent operated in this structure. As illustrated in Fig. 3, inertial navigation component and GNSS component are loosely coupled. If one component fails, the entire navigation system still outputs navigation information using the other navigation component [15]. This increases the reliability and robustness of the integrated system.

Tightly coupled integration architecture treats inertial sensors and GNSS as input sensors [1]. Integration filter estimates and output position estimates, velocity estimates and attitude estimates [1]. These estimates are feedback to GNSS. Fig. 4 illustrates the architecture of tightly coupled integration.

Tightly coupled architecture is more robust than loosely coupled architecture [1]. The integration filter feedbacks the errors estimates to the aiding satellite positioning system. This increases the accuracy of the system positioning results. It is more expensive to implement for system engineers [1] and requires more investment from business owner. Project manager would make more efforts in system development coordination. Tightly coupled architecture requires integrated navigation filter outputs to be fedback to satellite aiding system for correction. Since the algorithms of the satellite positioning systems developed by suppliers have been treated as black box for market users, this would add more difficulties in development of aiding algorithms for system engineers [1]. Therefore this architecture would cost more in investment, management and development. System engineers would make more design efforts in aiding and feedback algorithms for satellite positioning system correction module.

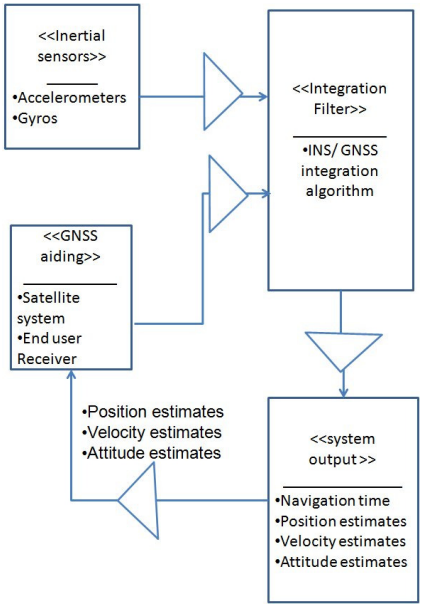

Fig. 4. Tightly coupled architecture

In our approach, we design a tool using a traceability matrix to analyze the characteristics, quality attributes and development constraints of considered integration architectures. Table III demonstrates some sample characteristics of each architecture pattern and their mapping to quality attributes and development constraints.

TABLE III. MAPPING INTEGRATED NAVIGATION SYSTEM ARCHITECTURE CHARACTERISTICS TO QUALITY ATTRIBUTES AND CONSTRAINTS

Architecture	**Characteristics (sample)**	**quality attributes and development constraints (sample)**
Architecture I: Direct feedforward architecture	Highly modular implementation	• Maintainability (high)
	High sampling rate, increase information availability	• Usability (high)
	Requires high process performance of hardware	• Hardware Cost (high)
	Positioning errors grow unbounded. Low accuracy in long term.	• Accuracy (medium)
	System continually outputs navigation information if one navigation component fails.	• Reliability (high) • Robustness (high)
Architecture II: Tightly coupled architecture	High investment in development	• Development Cost (high)
	Expensive to implement	• Development Cost (high)
	More efforts in collaboration with satellite system supplier	• Development Cost (high) • Development Time (long)_
	Difficult to implement	• Maintainability (low) • Development Cost (high)
		• Development Time (long)
	More efforts in development satellite aiding feedback algorithms	• Need for client collaboration capacity (high) • Development Cost (high) • Development Time (long)
	High accuracy for final integrated system	• Accuracy (high)
	More robust	• Robustness (high)
	Lower modular in maintenance	• Maintainability (low)

B. *Navigation architecture evaluation and decision*

After analysis of integration architecture pattern characteristics, and mapping them into quality attributes and development constraints, these analysis outcomes will be presented to stakeholders to evaluate.

Integration patterns are in standard form, some stakeholder requirements may not be within the standard form. In such situations, to meet all users' requirements in system specifications, further customized design is required in later stage. For example, security attribute is not addressed in these architectures. Another security module will be designed on top of the core integration layer to meet end users' requirements.

To assist stakeholder evaluation, in our approach a table form is designed for each stakeholder to provide a score against quality attributes and development constraints. Total evaluation score for each architecture pattern is calculated by applying a weight for each stakeholder. In a project development, each type of stakeholder's viewpoint should have different economic values at certain time frame. Table IV is the tool for architecture pattern evaluation. The final evaluation for each architecture pattern is calculated using Eq. 1.

$$Score(Arch_i) = weight(stakeholder) \times [\sum Score(stakeholder)] \quad (1)$$

where *wight*(*stakeholder*) is the weight of each stakeholder's score assigned by business owner and project manager. *Score*(*stakeholder*) is the evaluation score provided by each stakeholder against each quality attributes and development constraint. $\sum Score(stakeholder)$ is the summation of *Score*(*stakeholder*) for all quality attributes and constraints. *Score*(*Archi*) is the final total evaluation result for each architecture pattern *i* (i=I, II) by calculation in Eq. 1. Architecture decision is made using the evaluation results from Table IV and Eq. 1.

V. CONCLUSION AND FUTURE DIRECTION

In this paper, we present a development methodology for vehicle navigation systems. In this methodology an architectural prerogatives phase is designed in early development stage. Conventional plan-driven process starts from specification first then flows down to design. In this

research, an architectural prerogatives phase is designed before specifications. The aim of this phase is to provide a vision of system architecture for all developers before the start of detailed specifications. The architecture solution is derived from all stakeholder concerns. This makes the final system more supportable by stakeholders.

For the future research, more sensor integration architecture patterns will be studied. Other functional modules will be analyzed to make the final product more supportable by all stakeholder concerns.

REFERENCES

[1] S. Sukkarieh, Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles. PhD thesis, The University of Sydney, 2000

[2] X. Kong, L., Liu & H. G. Ryu. 'Activity-Artifact Flow of GPS/INS Integration for Positioning Error De-Correlation' in Fouzia Boukour Elbahhar and Atika Rivenq (eds), New Approach of Indoor and Outdoor Localization Systems, InTech, Rijeka, Crotia, pp. 193-212.

[3] R. Pressman, Software Engineering: A Practitioners Approach, 7e McGraw Hill 2010

[4] L. Guo, X. Kong, K. Sandrasegaran and X. Li Requirement-Design Framework of Integrated Navigation Systems, In Press

[5] L. Maciaszek. Requirements analysis and system design. Pearson Education. 2007

[6] G. Alfredo and A. Tundis. "A model-based method for system reliability analysis." Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium. Society for Computer Simulation International, 2012.

[7] B. Katherine A., and M.W. Sawyer. "Understanding the Human Component of Area Navigation Procedures Across the National Airspace System." Proceedings of the Human Factors and Ergonomics Society Annual Meeting. Vol. 57. No. 1. SAGE Publications, 2013.

[8] Bruna, O., et al. "Small Aircraft Emergency Landing Decision Support System–Pilots' performance Assessment." XX IMEKO World Congress, Busan, Republic of Korea. 2012.

[9] S. Krishna, et al. "Future e-enabled aircraft communications and security: The next 20 years and beyond." Proceedings of the IEEE 99.11 (2011): 2040-2055.

[10] L. Siegel and B. Malcolm "A field usability evaluation of a wearable system." Wearable Computers, 1997. Digest of Papers., First International Symposium on. IEEE, 1997.

[11] C. Coral, J. Ruiz, and M. Piattini "A web metrics survey using WQM." Web Engineering. Springer Berlin Heidelberg, 2004. 147-160.

[12] J. Reekie and J. Rohan, A software architecture primer. Angophora Press, 2006.

[13] Architecture_Working_Group, IEEE Std 1471-2000 Recommended Practice Architectural Description of Software-Intensive Systems, I.S.E.S. Committee, Editor. 2000, IEEE Standard 1471-2000:

[14] X. Li, L. Guo, X. Kong and S. Gao. Conflicts Analysis and Validation of Inertial Sensors Aided Global Positioning System Carrier Tracking Loop. Sensor Letters, 11(5), 805-811. 2013

[15] P. Maybeck. Stochastic models, estimation and control, New York, Academic Press, 1979

TABLE IV. INTEGRATED NAVIGATION SYSTEM ARCHITECTURE EVALUATION FORM

quality attributes and development constraints	**Architecture**		**Business owner Evaluation (Score)**		**Project manager Evaluation (Score)**		**System engineer Evaluation (Score)**		**System end user Evaluation (Score)**		**System maintainer Evaluation (Score)**	
	I	**II**										
	Direct feedforward	**Tightly coupled**	I	II	I	II	I	II	I	II	I	II
Accuracy	medium	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Robustness	medium	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Reliability	high	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Security	Not addressed	Not addressed	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Response time	low	Medium	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Maintainability	high	low	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Development complexity	low	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Cost constraints	low	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Time constraints	low	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Required supplier / client collaboration capacity	low	high	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Stakeholder evaluation score before weighting			[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Stakeholder weight			( )		( )		( )		( )		( )	
stakeholder evaluation score after weighting ∑			[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Total evaluation score	( )	( )										