A Case Study in Stakeholder-oriented Goal-modeling Framework

Jipeng Wu Eryu Ding Bin Luo

Software Institute
Nanjing University

ICSESS Presentations, 2014

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

- We present a requirements engineering framework SoF(Stakeholder-oriented goal-modeling Framework) based on RWS goal methods and KAOS modeling methods.
- This framework provides a consistent guide for requirements acquisition, requirements analysis and requirements validation, the three phases of requirements development.
- In the phase of requirements acquisition, SoF adopts structured text in nature language to generate a goal model from interview results. It is applied in the development of a management information system called SDMS for a technology department of a software company.

- We present a requirements engineering framework SoF(Stakeholder-oriented goal-modeling Framework) based on RWS goal methods and KAOS modeling methods.
- This framework provides a consistent guide for requirements acquisition, requirements analysis and requirements validation, the three phases of requirements development.
- In the phase of requirements acquisition, SoF adopts structured text in nature language to generate a goal model from interview results. It is applied in the development of a management information system called SDMS for a technology department of a software company.

- We present a requirements engineering framework SoF(Stakeholder-oriented goal-modeling Framework) based on RWS goal methods and KAOS modeling methods.
- This framework provides a consistent guide for requirements acquisition, requirements analysis and requirements validation, the three phases of requirements development.
- In the phase of requirements acquisition, SoF adopts structured text in nature language to generate a goal model from interview results. It is applied in the development of a management information system called SDMS for a technology department of a software company.

- We present a requirements engineering framework SoF(Stakeholder-oriented goal-modeling Framework) based on RWS goal methods and KAOS modeling methods.
- This framework provides a consistent guide for requirements acquisition, requirements analysis and requirements validation, the three phases of requirements development.
- In the phase of requirements acquisition, SoF adopts structured text in nature language to generate a goal model from interview results. It is applied in the development of a management information system called SDMS for a technology department of a software company.

- Goal methods now available are usually limited in a certain phase of requirements development and rely on some specific context. Considering the overall use of goal methods in requirements development, a goal-modeling framework applied throughout different phases of requirements development is necessary.
- SoF is not limited in a particular abstraction level. It is a method that traces the practical process of requirement acquisition, requirement analysis and requirement validation.
- In addition, current goal frameworks are more focused on analysis and technical details of goal modeling of some phase of requirements development. SoF, however, has a different focus more concentrated on how to use goal models to communicate with stakeholders and acquire feedback more agilely.

- Goal methods now available are usually limited in a certain phase of requirements development and rely on some specific context. Considering the overall use of goal methods in requirements development, a goal-modeling framework applied throughout different phases of requirements development is necessary.
- SoF is not limited in a particular abstraction level. It is a method that traces the practical process of requirement acquisition, requirement analysis and requirement validation.
- In addition, current goal frameworks are more focused on analysis and technical details of goal modeling of some phase of requirements development. SoF, however, has a different focus more concentrated on how to use goal models to communicate with stakeholders and acquire feedback more agilely.

- Goal methods now available are usually limited in a certain phase of requirements development and rely on some specific context. Considering the overall use of goal methods in requirements development, a goal-modeling framework applied throughout different phases of requirements development is necessary.
- SoF is not limited in a particular abstraction level. It is a method that traces the practical process of requirement acquisition, requirement analysis and requirement validation.
- In addition, current goal frameworks are more focused on analysis and technical details of goal modeling of some phase of requirements development. SoF, however, has a different focus more concentrated on how to use goal models to communicate with stakeholders and acquire feedback more agilely.

- Goal methods now available are usually limited in a certain phase of requirements development and rely on some specific context. Considering the overall use of goal methods in requirements development, a goal-modeling framework applied throughout different phases of requirements development is necessary.
- SoF is not limited in a particular abstraction level. It is a method that traces the practical process of requirement acquisition, requirement analysis and requirement validation.
- In addition, current goal frameworks are more focused on analysis and technical details of goal modeling of some phase of requirements development. SoF, however, has a different focus more concentrated on how to use goal models to communicate with stakeholders and acquire feedback more agilely.

KAOS

- The SoF goal-modeling is based on KAOS methods.
  1. KAOS provides a formal description of goal model.
  2. It defines some meta-conceptsgoal, action, agent, entity and event, which can be visualized as nodes.
  3. The edges between nodes capture the semantic links between such abstractions.
  4. Two basic link types are definedAND/OR.
- Van Lamsweerde extended the KAOS visualization with the following types of links: Contributes(+), ContributesStrongly(++), Conflicts(-), and ConflictsStrongly(–). Elicitation of goal model is implemented by asking How/Why questions.
- The deficiency of the practical use of KAOS methods is that it relies on a specific context of requirements development. SoF applies KAOS methods in concrete development process and offers its context.

KAOS

- The SoF goal-modeling is based on KAOS methods.
  1. KAOS provides a formal description of goal model.
  2. It defines some meta-conceptsgoal, action, agent, entity and event, which can be visualized as nodes.
  3. The edges between nodes capture the semantic links between such abstractions.
  4. Two basic link types are definedAND/OR.
- Van Lamsweerde extended the KAOS visualization with the following types of links: Contributes(+), ContributesStrongly(++), Conflicts(-), and ConflictsStrongly(–). Elicitation of goal model is implemented by asking How/Why questions.
- The deficiency of the practical use of KAOS methods is that it relies on a specific context of requirements development. SoF applies KAOS methods in concrete development process and offers its context.

KAOS

- The SoF goal-modeling is based on KAOS methods.
  1. KAOS provides a formal description of goal model.
  2. It defines some meta-conceptsgoal, action, agent, entity and event, which can be visualized as nodes.
  3. The edges between nodes capture the semantic links between such abstractions.
  4. Two basic link types are definedAND/OR.
- Van Lamsweerde extended the KAOS visualization with the following types of links: Contributes(+), ContributesStrongly(++), Conflicts(-), and ConflictsStrongly(–). Elicitation of goal model is implemented by asking How/Why questions.
- The deficiency of the practical use of KAOS methods is that it relies on a specific context of requirements development. SoF applies KAOS methods in concrete development process and offers its context.

KAOS

- The SoF goal-modeling is based on KAOS methods.
  1. KAOS provides a formal description of goal model.
  2. It defines some meta-conceptsgoal, action, agent, entity and event, which can be visualized as nodes.
  3. The edges between nodes capture the semantic links between such abstractions.
  4. Two basic link types are definedAND/OR.
- Van Lamsweerde extended the KAOS visualization with the following types of links: Contributes(+), ContributesStrongly(++), Conflicts(-), and ConflictsStrongly(–). Elicitation of goal model is implemented by asking How/Why questions.
- The deficiency of the practical use of KAOS methods is that it relies on a specific context of requirements development. SoF applies KAOS methods in concrete development process and offers its context.

KAOS

- The SoF goal-modeling is based on KAOS methods.
    1. KAOS provides a formal description of goal model.
    2. It defines some meta-conceptsgoal, action, agent, entity and event, which can be visualized as nodes.
    3. The edges between nodes capture the semantic links between such abstractions.
    4. Two basic link types are definedAND/OR.
- Van Lamsweerde extended the KAOS visualization with the following types of links: Contributes(+), ContributesStrongly(++), Conflicts(-), and ConflictsStrongly(–). Elicitation of goal model is implemented by asking How/Why questions.
- The deficiency of the practical use of KAOS methods is that it relies on a specific context of requirements development. SoF applies KAOS methods in concrete development process and offers its context.

RWS

- RWS annotated the goal model with views of stakeholders(1.agree, 2.not agree, 3.add more goals and 4.no position), which facilitates review and validation and finally conforms the goal model to the real world scene.
  1. RWS requires that current system should be captured in the form of rich media(e.g., taking photos, recording videos) and the observation results should be linked to goals, in order to elaborate and validate goals in the follow-up work. The observation results are called Real World Scene.
  2. The RWS approach is focused on goal analysis. It uses rich-media scene materials as basis for adding new goals and elaborating goals. The requirements are validated by requirements developers and domain experts. However the scene materials cannot capture full-scale requirements as observation results in requirements development. Additionally, RWS cannot ensure that the decisions made by requirements developers and domain experts

RWS

- RWS annotated the goal model with views of stakeholders(1.agree, 2.not agree, 3.add more goals and 4.no position), which facilitates review and validation and finally conforms the goal model to the real world scene.
  1. RWS requires that current system should be captured in the form of rich media(e.g., taking photos, recording videos) and the observation results should be linked to goals, in order to elaborate and validate goals in the follow-up work. The observation results are called Real World Scene.
  2. The RWS approach is focused on goal analysis. It uses rich-media scene materials as basis for adding new goals and elaborating goals. The requirements are validated by requirements developers and domain experts. However the scene materials cannot capture full-scale requirements as observation results in requirements development. Additionally, RWS cannot ensure that the decisions made by requirements developers and domain experts

RWS

- RWS annotated the goal model with views of stakeholders(1.agree, 2.not agree, 3.add more goals and 4.no position), which facilitates review and validation and finally conforms the goal model to the real world scene.
  1. RWS requires that current system should be captured in the form of rich media(e.g., taking photos, recording videos) and the observation results should be linked to goals, in order to elaborate and validate goals in the follow-up work. The observation results are called Real World Scene.
  2. The RWS approach is focused on goal analysis. It uses rich-media scene materials as basis for adding new goals and elaborating goals. The requirements are validated by requirements developers and domain experts. However the scene materials cannot capture full-scale requirements as observation results in requirements development. Additionally, RWS cannot ensure that the decisions made by requirements developers and domain experts

RWS

- RWS annotated the goal model with views of stakeholders(1.agree, 2.not agree, 3.add more goals and 4.no position), which facilitates review and validation and finally conforms the goal model to the real world scene.
    1. RWS requires that current system should be captured in the form of rich media(e.g., taking photos, recording videos) and the observation results should be linked to goals, in order to elaborate and validate goals in the follow-up work. The observation results are called Real World Scene.
    2. The RWS approach is focused on goal analysis. It uses rich-media scene materials as basis for adding new goals and elaborating goals. The requirements are validated by requirements developers and domain experts. However the scene materials cannot capture full-scale requirements as observation results in requirements development. Additionally, RWS cannot ensure that the decisions made by requirements developers and domain experts

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

Introduction

- SoF is a framework guiding the acquition of user requirements.
- SoF proposed a method ensuring that the goals correspond to stakeholders' interest and their authentic expectation based on RWS: The parent of a goal must have passed the stakeholder validation everytime before the goal is elaborated.
- SoF combines requirements acquition, requirements analysis and requirements validation as one atomic activity, which is called *SoF Elaboration Activity*. Each successful *SoF Elaboration Activity* includes the following phases:
  1. listening to stakeholders' opions and update structured text document
  2. elaborating goal models and solve conflicts by means of KAOS goal-modeling
  3. and finally convening stakeholder validation interview

Introduction

- SoF is a framework guiding the acquition of user requirements.
- SoF proposed a method ensuring that the goals correspond to stakeholders' interest and their authentic expectation based on RWS: The parent of a goal must have passed the stakeholder validation everytime before the goal is elaborated.
- SoF combines requirements acquition, requirements analysis and requirements validation as one atomic activity, which is called *SoF Elaboration Activity*. Each successful *SoF Elaboration Activity* includes the following phases:
  1. listening to stakeholders' opions and update structured text document
  2. elaborating goal models and solve conflicts by means of KAOS goal-modeling
  3. and finally convening stakeholder validation interview

Introduction

- SoF is a framework guiding the acquition of user requirements.
- SoF proposed a method ensuring that the goals correspond to stakeholders' interest and their authentic expectation based on RWS: The parent of a goal must have passed the stakeholder validation everytime before the goal is elaborated.
- SoF combines requirements acquition, requirements analysis and requirements validation as one atomic activity, which is called *SoF Elaboration Activity*. Each successful *SoF Elaboration Activity* includes the following phases:
  1. listening to stakeholders' opions and update structured text document
  2. elaborating goal models and solve conflicts by means of KAOS goal-modeling
  3. and finally convening stakeholder validation interview

Introduction

- If a requirements of the future system need *n* times of elaboration, its *SoF Elaboration Activity* should be conducted at least *n* times.
- Before all the steps of the *SoF Elaboration Activity* of one requirement have been finished, it is not allowed that the *SoF Elaboration Activity* of another requirement is initiated.

Introduction

- If a requirements of the future system need n times of elaboration, its *SoF Elaboration Activity* should be conducted at least n times.
- Before all the steps of the *SoF Elaboration Activity* of one requirement have been finished, it is not allowed that the *SoF Elaboration Activity* of another requirement is initiated.

Introduction

- If a requirements of the future system need *n* times of elaboration, its *SoF Elaboration Activity* should be conducted at least *n* times.
- Before all the steps of the *SoF Elaboration Activity* of one requirement have been finished, it is not allowed that the *SoF Elaboration Activity* of another requirement is initiated.

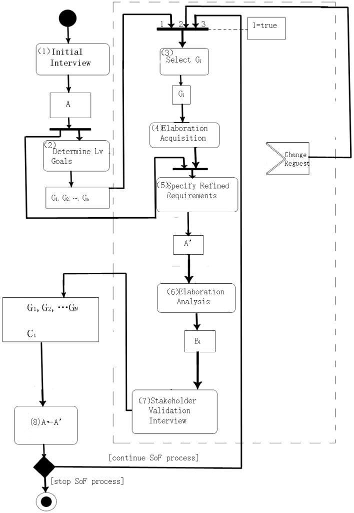

Figure : Activity Diagram of SoF Process

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact *A*. *A* is a structured text in nature language based on scenario description.
- To analyze *A* and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to *A* the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact *A*. *A* is a structured text in nature language based on scenario description.
- To analyze *A* and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to A the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact A. A is a structured text in nature language based on scenario description.
- To analyze A and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to A the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact *A*. *A* is a structured text in nature language based on scenario description.
- To analyze *A* and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to *A* the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact A. A is a structured text in nature language based on scenario description.
- To analyze A and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to A the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To convene an initial interview, and yield a requirements artefact A. A is a structured text in nature language based on scenario description.
- To analyze A and generate high-level goals: $G_1, G_2, \ldots, G_i, \ldots, G_n (i \in N^+)$.
- To select one leaf-node goal G_i.(If this goal G_i is not a high-level goal, the subscript i is a material path, e.g., the first child goal of G_1 is $G_{1.1}$).
- To ask How/Why questions about G_i to acquire stakeholders' viewpoints for requirements elaboration.
- To generate A' by adding to A the stakeholders' description of elaborated child goals.

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To yield a requirements artefact B_i with information from $A' \cup A$ and knowledge of requirements developers and domain experts. B_i is a SoF annotated goal tree model describing the future system.
- To convene a stakeholder interview to validate the requirements artefact B_i with the newest version of A_i as a specification of collected stakeholders' viewpoints, and finally yield a requirements artefact C_i. C_i is a SoF annotated tree that has passed the validation. New goals in C_i are marked with $G_{i.j}(j \in N^+)$.
- To rename the temporary A' to A as a formal version. (After this step, the requirements developer should decide whether or not to end the SoF process. To continue SoF process, jump to Step 3 and select another goal.)

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To yield a requirements artefact B_i with information from $A' \cup A$ and knowledge of requirements developers and domain experts. B_i is a SoF annotated goal tree model describing the future system.
- To convene a stakeholder interview to validate the requirements artefact B_i with the newest version of A_i as a specification of collected stakeholders' viewpoints, and finally yield a requirements artefact C_i. C_i is a SoF annotated tree that has passed the validation. New goals in C_i are marked with $G_{i.j}(j \in N^+)$.
- To rename the temporary A' to A as a formal version. (After this step, the requirements developer should decide whether or not to end the SoF process. To continue SoF process, jump to Step 3 and select another goal.)

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To yield a requirements artefact B_i with information from $A' \cup A$ and knowledge of requirements developers and domain experts. B_i is a SoF annotated goal tree model describing the future system.
- To convene a stakeholder interview to validate the requirements artefact B_i with the newest version of A_i as a specification of collected stakeholders' viewpoints, and finally yield a requirements artefact C_i. C_i is a SoF annotated tree that has passed the validation. New goals in C_i are marked with $G_{i.j}(j \in N^+)$.
- To rename the temporary A' to A as a formal version. (After this step, the requirements developer should decide whether or not to end the SoF process. To continue SoF process, jump to Step 3 and select another goal.)

As is shown in Fig.1, the 8 detailed steps of SoF is as follows:

- To yield a requirements artefact B_i with information from $A' \cup A$ and knowledge of requirements developers and domain experts. B_i is a SoF annotated goal tree model describing the future system.
- To convene a stakeholder interview to validate the requirements artefact B_i with the newest version of A_i as a specification of collected stakeholders' viewpoints, and finally yield a requirements artefact C_i. C_i is a SoF annotated tree that has passed the validation. New goals in C_i are marked with $G_{i.j}(j \in N^+)$.
- To rename the temporary A' to A as a formal version. (After this step, the requirements developer should decide whether or not to end the SoF process. To continue SoF process, jump to Step 3 and select another goal.)

Other Features

- SoF introduces a mechanism against requirements change. It allows stakeholders to send change requests at any time in the SoF process, which is the only legal way to interrupt SoF Elaboration Activities.
- The final artefact of SoF is a goal model whose elaboration has passed stakeholder validation. This model is well refined and accepted by stakeholders. It can be a reliable description of user requirements.

Other Features

- SoF introduces a mechanism against requirements change. It allows stakeholders to send change requests at any time in the SoF process, which is the only legal way to interrupt SoF Elaboration Activities.
- The final artefact of SoF is a goal model whose elaboration has passed stakeholder validation. This model is well refined and accepted by stakeholders. It can be a reliable description of user requirements.

Other Features

- SoF introduces a mechanism against requirements change. It allows stakeholders to send change requests at any time in the SoF process, which is the only legal way to interrupt SoF Elaboration Activities.
- The final artefact of SoF is a goal model whose elaboration has passed stakeholder validation. This model is well refined and accepted by stakeholders. It can be a reliable description of user requirements.

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

- SoF adopts a traditional way, conducting a interview, to acquire requirements. It is insufficient to rely only on stakeholders' knowledge and judgment. Thus it is necessary to find a way to specify the description of scenarios of future system from stakeholders and help stakeholders to obtain a consensus.
- SoF uses a structured text in nature language based on scenarios to organize interview results, recording stakeholders' expectations of future system.
- The initial interview generates a brief structured specification, describing high-level goals in the form of scenarios. This specification is updated after requirements acquisition phase of each *SoF Elaboration Activity*. Each update is a basis of subsequent goal refinement and goal validation.
- The specification is maintained by the requirements development team. They are responsible to update the specification according to change decisions made by stakeholders. Scenarios are used to describe the conceptual model of a future system. The questions posed in stakeholder interviews are based on different scenarios in order to solicit and specify their needs and expectations.

- SoF adopts a traditional way, conducting a interview, to acquire requirements. It is insufficient to rely only on stakeholders' knowledge and judgment. Thus it is necessary to find a way to specify the description of scenarios of future system from stakeholders and help stakeholders to obtain a consensus.
- SoF uses a structured text in nature language based on scenarios to organize interview results, recording stakeholders' expectations of future system.
- The initial interview generates a brief structured specification, describing high-level goals in the form of scenarios. This specification is updated after requirements acquisition phase of each *SoF Elaboration Activity*. Each update is a basis of subsequent goal refinement and goal validation.
- The specification is maintained by the requirements development team. They are responsible to update the specification according to change decisions made by stakeholders. Scenarios are used to describe the conceptual model of a future system. The questions posed in stakeholder interviews are based on different scenarios in order to solicit and specify their needs and expectations.

- SoF adopts a traditional way, conducting a interview, to acquire requirements. It is insufficient to rely only on stakeholders' knowledge and judgment. Thus it is necessary to find a way to specify the description of scenarios of future system from stakeholders and help stakeholders to obtain a consensus.
- SoF uses a structured text in nature language based on scenarios to organize interview results, recording stakeholders' expectations of future system.
- The initial interview generates a brief structured specification, describing high-level goals in the form of scenarios. This specification is updated after requirements acquisition phase of each *SoF Elaboration Activity*. Each update is a basis of subsequent goal refinement and goal validation.
- The specification is maintained by the requirements development team. They are responsible to update the specification according to change decisions made by stakeholders. Scenarios are used to describe the conceptual model of a future system. The questions posed in stakeholder interviews are based on different scenarios in order to solicit and specify their needs and expectations.

- SoF adopts a traditional way, conducting a interview, to acquire requirements. It is insufficient to rely only on stakeholders' knowledge and judgment. Thus it is necessary to find a way to specify the description of scenarios of future system from stakeholders and help stakeholders to obtain a consensus.
- SoF uses a structured text in nature language based on scenarios to organize interview results, recording stakeholders' expectations of future system.
- The initial interview generates a brief structured specification, describing high-level goals in the form of scenarios. This specification is updated after requirements acquisition phase of each *SoF Elaboration Activity*. Each update is a basis of subsequent goal refinement and goal validation.
- The specification is maintained by the requirements development team. They are responsible to update the specification according to change decisions made by stakeholders. Scenarios are used to describe the conceptual model of a future system. The questions posed in stakeholder interviews are based on different scenarios in order to solicit and specify their needs and expectations.

- SoF adopts a traditional way, conducting a interview, to acquire requirements. It is insufficient to rely only on stakeholders' knowledge and judgment. Thus it is necessary to find a way to specify the description of scenarios of future system from stakeholders and help stakeholders to obtain a consensus.
- SoF uses a structured text in nature language based on scenarios to organize interview results, recording stakeholders' expectations of future system.
- The initial interview generates a brief structured specification, describing high-level goals in the form of scenarios. This specification is updated after requirements acquisition phase of each *SoF Elaboration Activity*. Each update is a basis of subsequent goal refinement and goal validation.
- The specification is maintained by the requirements development team. They are responsible to update the specification according to change decisions made by stakeholders. Scenarios are used to describe the conceptual model of a future system. The questions posed in stakeholder interviews are based on different scenarios in order to solicit and specify their needs and expectations.

Outline

1 Abstract

2 Introduction

3 Related Work

4 Stakeholder-oriented Goal-modeling FrameworkSoF
- SoF Process
- Structured Text in Nature Language applied in SoF
- SoF Annotated Goal Tree Model
- Scope of SoF

5 SoF Case Study
- Background of SDMS
- Structured Specification of StakeholdersPerspectives
- SoF Annotated Goal Tree Model

6 Discussion

- The goal model adopted by SoF is named after *SoF Annotated Goal Tree Model*. It is based on KAOS and extended by introducing annotations for stakeholder evaluation.
- *SoF Annotated Goal Tree Model* is a KAOS goal model which is top-down decomposed. SoF adopts methods from RWS to evaluate requirements with the help of annotated goal models. Only the goals that have passed stakeholder evaluation are allowed to be a basis of elaboration.
- The annotations can be used to organize more structured evaluation interviews and actively engage stakeholders in such interviews. More comprehensive engagement among stakeholders allow requirements developers to understand their negotiation, mutual understanding, divergence and consensus of a conceptual future system. After high-level goals determined in an initial interview, the subsequent activities, such as goal elicitation, conflict resolution and requirements evaluation, will be executed based on *SoF Annotated Goal Tree Model*.
- More detailed introduction will be presented in the following case study.

- The goal model adopted by SoF is named after *SoF Annotated Goal Tree Model*. It is based on KAOS and extended by introducing annotations for stakeholder evaluation.
- *SoF Annotated Goal Tree Model* is a KAOS goal model which is top-down decomposed. SoF adopts methods from RWS to evaluate requirements with the help of annotated goal models. Only the goals that have passed stakeholder evaluation are allowed to be a basis of elaboration.
- The annotations can be used to organize more structured evaluation interviews and actively engage stakeholders in such interviews. More comprehensive engagement among stakeholders allow requirements developers to understand their negotiation, mutual understanding, divergence and consensus of a conceptual future system. After high-level goals determined in an initial interview, the subsequent activities, such as goal elicitation, conflict resolution and requirements evaluation, will be executed based on *SoF Annotated Goal Tree Model*.
- More detailed introduction will be presented in the following case study.

- The goal model adopted by SoF is named after *SoF Annotated Goal Tree Model*. It is based on KAOS and extended by introducing annotations for stakeholder evaluation.
- *SoF Annotated Goal Tree Model* is a KAOS goal model which is top-down decomposed. SoF adopts methods from RWS to evaluate requirements with the help of annotated goal models. Only the goals that have passed stakeholder evaluation are allowed to be a basis of elaboration.
- The annotations can be used to organize more structured evaluation interviews and actively engage stakeholders in such interviews. More comprehensive engagement among stakeholders allow requirements developers to understand their negotiation, mutual understanding, divergence and consensus of a conceptual future system. After high-level goals determined in an initial interview, the subsequent activities, such as goal elicitation, conflict resolution and requirements evaluation, will be executed based on *SoF Annotated Goal Tree Model*.
- More detailed introduction will be presented in the following case study.

- The goal model adopted by SoF is named after *SoF Annotated Goal Tree Model*. It is based on KAOS and extended by introducing annotations for stakeholder evaluation.
- *SoF Annotated Goal Tree Model* is a KAOS goal model which is top-down decomposed. SoF adopts methods from RWS to evaluate requirements with the help of annotated goal models. Only the goals that have passed stakeholder evaluation are allowed to be a basis of elaboration.
- The annotations can be used to organize more structured evaluation interviews and actively engage stakeholders in such interviews. More comprehensive engagement among stakeholders allow requirements developers to understand their negotiation, mutual understanding, divergence and consensus of a conceptual future system. After high-level goals determined in an initial interview, the subsequent activities, such as goal elicitation, conflict resolution and requirements evaluation, will be executed based on *SoF Annotated Goal Tree Model*.
- More detailed introduction will be presented in the following case study.

Outline

1 Abstract

2 Introduction

3 Related Work

4 Stakeholder-oriented Goal-modeling FrameworkSoF
- SoF Process
- Structured Text in Nature Language applied in SoF
- SoF Annotated Goal Tree Model
- Scope of SoF

5 SoF Case Study
- Background of SDMS
- Structured Specification of StakeholdersPerspectives
- SoF Annotated Goal Tree Model

6 Discussion

- SoF is not only a method to define goal models. It applies goal models to guide early phase of requirements development.
- SoF method is not exclusive. It can be used as supplement of traditional use case model and object-oriented methods.
- The primary focus of SoF is the early phase of requirements development. Further refinement to system-level requirements should rely on analysis using other methods, e.g., object-oriented methods.

- SoF is not only a method to define goal models. It applies goal models to guide early phase of requirements development.
- SoF method is not exclusive. It can be used as supplement of traditional use case model and object-oriented methods.
- The primary focus of SoF is the early phase of requirements development. Further refinement to system-level requirements should rely on analysis using other methods, e.g., object-oriented methods.

- SoF is not only a method to define goal models. It applies goal models to guide early phase of requirements development.
- SoF method is not exclusive. It can be used as supplement of traditional use case model and object-oriented methods.
- The primary focus of SoF is the early phase of requirements development. Further refinement to system-level requirements should rely on analysis using other methods, e.g., object-oriented methods.

- SoF is not only a method to define goal models. It applies goal models to guide early phase of requirements development.
- SoF method is not exclusive. It can be used as supplement of traditional use case model and object-oriented methods.
- The primary focus of SoF is the early phase of requirements development. Further refinement to system-level requirements should rely on analysis using other methods, e.g., object-oriented methods.

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

- SDMS(Software Departmental Management System) is an information system for the technical department of a software company aimed at improving their efficiency.

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

- The initial interview poses questions about the background, problem domain, workflow of all possible scenarios of the future system. Requirements developers are responsible for combining information fragments obtained from the interview together to generate a structured specification.
- In the subsequent *SoF Elaboration Activities*, more details will be added in the structured specification in nature language. *SoF Annotated Goal Tree Model* and the structured specification based on scenarios are continuously ameliorated when top-down *SoF Elaboration Activities* are carried out gradually. After SoF process, these requirements artefacts can reach the same level of completeness.
- Fig.2 presents a structured description of a scenario of SDMS. It specifies possible events, trigger events of these events and concrete actions of stakeholders in these events in nature languge. Fig.2 does not present a cursory description of high-level goals. It is a specification of a scenario which have passed one *SoF Elaboration Activity* and have been enriched with some details.

- The initial interview poses questions about the background, problem domain, workflow of all possible scenarios of the future system. Requirements developers are responsible for combining information fragments obtained from the interview together to generate a structured specification.
- In the subsequent *SoF Elaboration Activities*, more details will be added in the structured specification in nature language. *SoF Annotated Goal Tree Model* and the structured specification based on scenarios are continuously ameliorated when top-down *SoF Elaboration Activities* are carried out gradually. After SoF process, these requirements artefacts can reach the same level of completeness.
- Fig.2 presents a structured description of a scenario of SDMS. It specifies possible events, trigger events of these events and concrete actions of stakeholders in these events in nature languge. Fig.2 does not present a cursory description of high-level goals. It is a specification of a scenario which have passed one *SoF Elaboration Activity* and have been enriched with some details.

- The initial interview poses questions about the background, problem domain, workflow of all possible scenarios of the future system. Requirements developers are responsible for combining information fragments obtained from the interview together to generate a structured specification.
- In the subsequent *SoF Elaboration Activities*, more details will be added in the structured specification in nature language. *SoF Annotated Goal Tree Model* and the structured specification based on scenarios are continuously ameliorated when top-down *SoF Elaboration Activities* are carried out gradually. After SoF process, these requirements artefacts can reach the same level of completeness.
- Fig.2 presents a structured description of a scenario of SDMS. It specifies possible events, trigger events of these events and concrete actions of stakeholders in these events in nature languge. Fig.2 does not present a cursory description of high-level goals. It is a specification of a scenario which have passed one *SoF Elaboration Activity* and have been enriched with some details.

- The initial interview poses questions about the background, problem domain, workflow of all possible scenarios of the future system. Requirements developers are responsible for combining information fragments obtained from the interview together to generate a structured specification.
- In the subsequent *SoF Elaboration Activities*, more details will be added in the structured specification in nature language. *SoF Annotated Goal Tree Model* and the structured specification based on scenarios are continuously ameliorated when top-down *SoF Elaboration Activities* are carried out gradually. After SoF process, these requirements artefacts can reach the same level of completeness.
- Fig.2 presents a structured description of a scenario of SDMS. It specifies possible events, trigger events of these events and concrete actions of stakeholders in these events in nature languge. Fig.2 does not present a cursory description of high-level goals. It is a specification of a scenario which have passed one *SoF Elaboration Activity* and have been enriched with some details.

```
SceneName: ProjectManagement

Concerns: Manager of Technology Department as A0, Project Manager as A1.

SceneDescription:

(Event 1) → EventName: Creating a new Project.

          → TriggerEvent: The company wins a bid.

          → Action: (1) A0 fills in the properties of the project.

                    (2) A0 confirms to save the new project data.

(Event 2) → EventName: Selecting Development Team.

         → TriggerEvent: Event 1

          → Action: (1) Before making a choice, A0 sholud look through inf-

                        ormation of developers to make sure that they are

                        properly-qualified.

                    (2) A0 selects project manager and developers from a

                        list provided by the future system, which creates

                        a new team.

                    (3) A0  confirms to save the data of a team responsible

                       for a project.

(Event 3) → EventName: Editting Project State

         → TriggerEvent: A periodic timeout event

         → Action: (1) A1 Edits the progress and other information of his

                      current project.

                   (2) A1 Confirms to save the new state of project.

(Event 4) → EventName: Querying Project State

         → TriggerEvent: A0 wants to know about current progress of a cert-

                   ain project.

         → Action: (1) A0 selects a project.
```

- In this example, stakeholders describe the scenario how a technical manager manage projects. This scenario includes four events:
    1. *create a project*
    2. *select development team*
    3. *edit project info*
    4. *and query project progress*
- Details of these events are specified in the form of triggering events and action description. Requirements developers can implement the step of goal elaboration in *SoF Elaboration Activities* based on this specification. For example, they can decompose the high-level goal, *improve efficiency of project management*, to several child goals supporting this goal.

- In this example, stakeholders describe the scenario how a technical manager manage projects. This scenario includes four events:
  1. *create a project*
  2. *select development team*
  3. *edit project info*
  4. *and query project progress*
- Details of these events are specified in the form of triggering events and action description. Requirements developers can implement the step of goal elaboration in *SoF Elaboration Activities* based on this specification. For example, they can decompose the high-level goal, *improve efficiency of project management*, to several child goals supporting this goal.

- The grammar and structure of such structured description can be defined by requirements developers. The main purpose is to organize readable documents from requirements fragments in accordance with their scenarios for the convenience of modification of subsequent work and requirements evaluation.
- This structured descriptive document does not require rigorously strict grammar. It should be informal and simple enough. At the highest level of abstraction of requirements, we do not require strict and exact formal statement. It is more important to improve the intelligibility of the specification and control the cost of documentation. The specification is used only as a basis of evaluation of goal models of the future system. However it is not used in any steps of requirements analysis, such as requirements elaboration and solving goal conflicts, which are in the charge of *SoF Annotated Goal Tree Model*.

- The grammar and structure of such structured description can be defined by requirements developers. The main purpose is to organize readable documents from requirements fragments in accordance with their scenarios for the convenience of modification of subsequent work and requirements evaluation.
- This structured descriptive document does not require rigorously strict grammar. It should be informal and simple enough. At the highest level of abstraction of requirements, we do not require strict and exact formal statement. It is more important to improve the intelligibility of the specification and control the cost of documentation. The specification is used only as a basis of evaluation of goal models of the future system. However it is not used in any steps of requirements analysis, such as requirements elaboration and solving goal conflicts, which are in the charge of *SoF Annotated Goal Tree Model*.

- The grammar and structure of such structured description can be defined by requirements developers. The main purpose is to organize readable documents from requirements fragments in accordance with their scenarios for the convenience of modification of subsequent work and requirements evaluation.
- This structured descriptive document does not require rigorously strict grammar. It should be informal and simple enough. At the highest level of abstraction of requirements, we do not require strict and exact formal statement. It is more important to improve the intelligibility of the specification and control the cost of documentation. The specification is used only as a basis of evaluation of goal models of the future system. However it is not used in any steps of requirements analysis, such as requirements elaboration and solving goal conflicts, which are in the charge of *SoF Annotated Goal Tree Model*.

Outline

1. Abstract
2. Introduction
3. Related Work
4. Stakeholder-oriented Goal-modeling FrameworkSoF
   - SoF Process
   - Structured Text in Nature Language applied in SoF
   - SoF Annotated Goal Tree Model
   - Scope of SoF
5. SoF Case Study
   - Background of SDMS
   - Structured Specification of StakeholdersPerspectives
   - SoF Annotated Goal Tree Model
6. Discussion

Introduction

- *SoF Annotated Goal Tree Model* that has not passed evaluation is a driven model for communication with stakeholders and control of requirements changes. *SoF Annotated Goal Tree Model* that has passed evaluation is a basis of the next *SoF Elaboration Activity*.
- We designed a comparatively simpler annotation in the case SDMS because there are no complex stakeholder constituents or serious interest conflicts. The annotation of *SoF Annotated Goal Tree Model* can be modified if more complex evaluation process design is required.
- Evaluation phase of *SoF Elaboration Activity* can be further decomposed according to the design of annotations of *SoF Annotated Goal Tree Model*. SDMS adopts two evaluation steps: relevance validation and success validation.

Introduction

- *SoF Annotated Goal Tree Model* that has not passed evaluation is a driven model for communication with stakeholders and control of requirements changes. *SoF Annotated Goal Tree Model* that has passed evaluation is a basis of the next *SoF Elaboration Activity*.
- We designed a comparatively simpler annotation in the case SDMS because there are no complex stakeholder constituents or serious interest conflicts. The annotation of *SoF Annotated Goal Tree Model* can be modified if more complex evaluation process design is required.
- Evaluation phase of *SoF Elaboration Activity* can be further decomposed according to the design of annotations of *SoF Annotated Goal Tree Model*. SDMS adopts two evaluation steps: relevance validation and success validation.

Introduction

- *SoF Annotated Goal Tree Model* that has not passed evaluation is a driven model for communication with stakeholders and control of requirements changes. *SoF Annotated Goal Tree Model* that has passed evaluation is a basis of the next *SoF Elaboration Activity*.
- We designed a comparatively simpler annotation in the case SDMS because there are no complex stakeholder constituents or serious interest conflicts. The annotation of *SoF Annotated Goal Tree Model* can be modified if more complex evaluation process design is required.
- Evaluation phase of *SoF Elaboration Activity* can be further decomposed according to the design of annotations of *SoF Annotated Goal Tree Model*. SDMS adopts two evaluation steps: relevance validation and success validation.

Introduction

- *SoF Annotated Goal Tree Model* that has not passed evaluation is a driven model for communication with stakeholders and control of requirements changes. *SoF Annotated Goal Tree Model* that has passed evaluation is a basis of the next *SoF Elaboration Activity*.
- We designed a comparatively simpler annotation in the case SDMS because there are no complex stakeholder constituents or serious interest conflicts. The annotation of *SoF Annotated Goal Tree Model* can be modified if more complex evaluation process design is required.
- Evaluation phase of *SoF Elaboration Activity* can be further decomposed according to the design of annotations of *SoF Annotated Goal Tree Model*. SDMS adopts two evaluation steps: relevance validation and success validation.

SoF Annotated Goal Tree Model adds the following two types of marks:

- The mark "relevance" is used to record whether a goal has passed relevance validation. The evaluation is based on stakeholders perspectives and latest structured specification based on scenarios. If the goal is irrelevant to the future system, it is marked with No, otherwise it is marked with Yes. Goals cannot enter the next step of success validation until it passes the relevance validation.
- The mark "agreed" is used to record whether a relevant goal has passed success validation. The evaluation is based on stakeholders perspectives on the practical significance, constraints and cost of the evaluated goal. If a it does not agree with stakeholders' expectations, the goal should be still denied.

SoF Annotated Goal Tree Model adds the following two types of marks:

- The mark "relevance" is used to record whether a goal has passed relevance validation. The evaluation is based on stakeholders perspectives and latest structured specification based on scenarios. If the goal is irrelevant to the future system, it is marked with No, otherwise it is marked with Yes. Goals cannot enter the next step of success validation until it passes the relevance validation.
- The mark "agreed" is used to record whether a relevant goal has passed success validation. The evaluation is based on stakeholders perspectives on the practical significance, constraints and cost of the evaluated goal. If a it does not agree with stakeholders' expectations, the goal should be still denied.

SoF Annotated Goal Tree Model adds the following two types of marks:

- The mark "relevance" is used to record whether a goal has passed relevance validation. The evaluation is based on stakeholders perspectives and latest structured specification based on scenarios. If the goal is irrelevant to the future system, it is marked with No, otherwise it is marked with Yes. Goals cannot enter the next step of success validation until it passes the relevance validation.
- The mark "agreed" is used to record whether a relevant goal has passed success validation. The evaluation is based on stakeholders perspectives on the practical significance, constraints and cost of the evaluated goal. If a it does not agree with stakeholders' expectations, the goal should be still denied.

Goal Elaboration

- The process of requirements elaboration is consistent with KAOS process. Requirements conflicts should be solved during elaboration processes. In addition, the precondition of elaboration of a goal is that it has passed both steps of evaluation.
- In practical modeling, all the high-level goals should be completed in one diagram. This paper only presents a elaboration process of one of the high-level goals.

Goal Elaboration

- The process of requirements elaboration is consistent with KAOS process. Requirements conflicts should be solved during elaboration processes. In addition, the precondition of elaboration of a goal is that it has passed both steps of evaluation.
- In practical modeling, all the high-level goals should be completed in one diagram. This paper only presents a elaboration process of one of the high-level goals.

Goal Elaboration

- The process of requirements elaboration is consistent with KAOS process. Requirements conflicts should be solved during elaboration processes. In addition, the precondition of elaboration of a goal is that it has passed both steps of evaluation.
- In practical modeling, all the high-level goals should be completed in one diagram. This paper only presents a elaboration process of one of the high-level goals.

Goal Elaboration

- 1. In Fig.3 we presents a high-level goal G_1"To make technical department manager make better decisions when build project teams".
  2. By asking Why/How questions we get child goals supporting G_1: $G_{1.1}$ and $G_{1.2}$.
     1. $G_{1.1}$ is "To provide better support of information on developers and project managers for techinical department manager".
     2. $G_{1.2}$ is "To provide a mechanism allowing developers to reply with feedback to the manager's decisions".
  3. These two goals were validated as child goals supporting their parent goal in different aspects, and thus successfully passed relevance validation. Stakeholders agreed that these two goals are consistent with their expectations. The goal tree was allowed to be further refined after each goal had passed both validations.

Goal Elaboration

- 1. In Fig.3 we presents a high-level goal G_1"To make technical department manager make better decisions when build project teams".
  2. By asking Why/How questions we get child goals supporting G_1: $G_{1.1}$ and $G_{1.2}$.
     1. $G_{1.1}$ is "To provide better support of information on developers and project managers for techinical department manager".
     2. $G_{1.2}$ is "To provide a mechanism allowing developers to reply with feedback to the manager's decisions".
  3. These two goals were validated as child goals supporting their parent goal in different aspects, and thus successfully passed relevance validation. Stakeholders agreed that these two goals are consistent with their expectations. The goal tree was allowed to be further refined after each goal had passed both validations.

Goal Elaboration

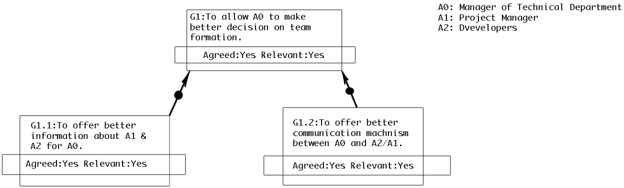
G1:To allow A0 to make better decision on team formation.
Agreed:Yes Relevant:Yes
G1.1:To offer better information about A1 & A2 for A0.
Agreed:Yes Relevant:Yes
G1.2:To offer better communication machnism between A0 and A2/A1.
Agreed:Yes Relevant:Yes
A0: Manager of Technical Department
A1: Project Manager
A2: Dvevelopers

Figure : SoF Annotated Goal Tree(ValidatedLevel 1 Elaboration)

Further Elaboration

- In Fig.4, we get a complete *SoF Annotated Goal Tree Model* that has been further elaborated. To emphasize the grammar, we omit detailed description of each goal in Fig.4 and represent each goal only with its indexed symbol.
- Here lists the detailed description of each goal that is not mentioned above.

  1. $G_{1.1.1}$ is "Project managers input information of developers".
  2. $G_{1.1.2}$ is "To access information of developers".
  3. $G_{1.1.3}$ is "Project managers should update information of developers periodically".
  4. $G_{1.1.1} G_{1.1.2} G_{1.1.3}$ work together to support their parent goal.

  If any of them fails to pass the evaluation, the whole elaboration plan fails.

Further Elaboration

- In Fig.4, we get a complete *SoF Annotated Goal Tree Model* that has been further elaborated. To emphasize the grammar, we omit detailed description of each goal in Fig.4 and represent each goal only with its indexed symbol.
- Here lists the detailed description of each goal that is not mentioned above.
  1. $G_{1.1.1}$ is "Project managers input information of developers".
  2. $G_{1.1.2}$ is "To access information of developers".
  3. $G_{1.1.3}$ is "Project managers should update information of developers periodically".
  4. $G_{1.1.1}G_{1.1.2}G_{1.1.3}$ work together to support their parent goal.

  If any of them fails to pass the evaluation, the whole elaboration plan fails.

Further Elaboration

- In Fig.4, we get a complete *SoF Annotated Goal Tree Model* that has been further elaborated. To emphasize the grammar, we omit detailed description of each goal in Fig.4 and represent each goal only with its indexed symbol.
- Here lists the detailed description of each goal that is not mentioned above.
  1. $G_{1.1.1}$ is "Project managers input information of developers".
  2. $G_{1.1.2}$ is "To access information of developers".
  3. $G_{1.1.3}$ is "Project managers should update information of developers periodically".
  4. $G_{1.1.1} G_{1.1.2} G_{1.1.3}$ work together to support their parent goal.

  If any of them fails to pass the evaluation, the whole elaboration plan fails.

Further Elaboration

- 1. $G_{1.2.1}$ is "To provide an instant messaging platform for technical department managers, project managers and developers".
  2. $G_{1.2.2}$ is "To publish decisions made by technical department managers and allow developers and project managers to reply asynchronously".
  3. $G_{1.2.1}$ does not conflict any other goal, and facilitates $G_{1.1}$ because the establishment of communication platform contributes positively to technical department managers' knowledge of developers' information. Thus $G_{1.2.1}$ passed both validations.
  4. $G_{1.2.2}$, although had passed the relevance validation, however, failed to pass the success validation because stakeholders believe that asynchronous communication is not practical and efficient enough to ensure the timeliness and richness of feedbacks.

Further Elaboration

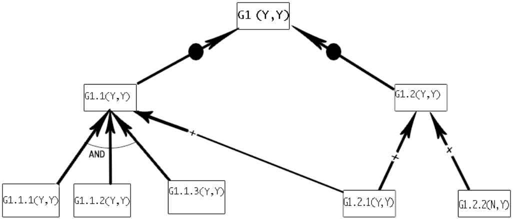

Figure : SoF Annotated Goal Tree(ValidatedLevel 2 Elaboration)

Conclusion:

- SoF can effectively acquire more exact requirements in practice of projects with nondeterminism of requirements. Each step of SoF activities is stakeholder-centered, which ensures that stakeholders' description of future system consists with goal models. With such consistency, SoF provides a reasonable context for KAOS application.
- *SoF Annotated Goal Tree Model* is an effective tool for not only analysis but also communication. In SDMS practice, structured nature language adopted by SoF is also proved an acceptable method to organize results of interviews, although its effectiveness and its cost control still require further validation from more practical cases.

Conclusion:

- SoF can effectively acquire more exact requirements in practice of projects with nondeterminism of requirements. Each step of SoF activities is stakeholder-centered, which ensures that stakeholders' description of future system consists with goal models. With such consistency, SoF provides a reasonable context for KAOS application.
- *SoF Annotated Goal Tree Model* is an effective tool for not only analysis but also communication. In SDMS practice, structured nature language adopted by SoF is also proved an acceptable method to organize results of interviews, although its effectiveness and its cost control still require further validation from more practical cases.

Conclusion:

- SoF can effectively acquire more exact requirements in practice of projects with nondeterminism of requirements. Each step of SoF activities is stakeholder-centered, which ensures that stakeholders' description of future system consists with goal models. With such consistency, SoF provides a reasonable context for KAOS application.
- *SoF Annotated Goal Tree Model* is an effective tool for not only analysis but also communication. In SDMS practice, structured nature language adopted by SoF is also proved an acceptable method to organize results of interviews, although its effectiveness and its cost control still require further validation from more practical cases.

Cost Problem:

1. The development cost is relatively higher because recursively executing SoF Elaboration Activities requires considerably frequent communication between and among requirements developers and stakeholders, which will certainly result in unacceptably high cost under some circumstances.
2. One of the most important sources of cost is the collection and management of structured specifications in nature language. We need to find a better way to process raw data from stakeholders.
3. Another source of cost is the ubiquitous involvement of stakeholders. We should decide whether a project is adaptive in applying SoF methods at the beginning of each project. Some projects with relatively constant and determinate requirements are not supposed to apply SoF methods.

Cost Problem:

1. The development cost is relatively higher because recursively executing SoF Elaboration Activities requires considerably frequent communication between and among requirements developers and stakeholders, which will certainly result in unacceptably high cost under some circumstances.
2. One of the most important sources of cost is the collection and management of structured specifications in nature language. We need to find a better way to process raw data from stakeholders.
3. Another source of cost is the ubiquitous involvement of stakeholders. We should decide whether a project is adaptive in applying SoF methods at the beginning of each project. Some projects with relatively constant and determinate requirements are not supposed to apply SoF methods.

Cost Problem:

1. The development cost is relatively higher because recursively executing SoF Elaboration Activities requires considerably frequent communication between and among requirements developers and stakeholders, which will certainly result in unacceptably high cost under some circumstances.
2. One of the most important sources of cost is the collection and management of structured specifications in nature language. We need to find a better way to process raw data from stakeholders.
3. Another source of cost is the ubiquitous involvement of stakeholders. We should decide whether a project is adaptive in applying SoF methods at the beginning of each project. Some projects with relatively constant and determinate requirements are not supposed to apply SoF methods.

Cost Problem:

1. The development cost is relatively higher because recursively executing SoF Elaboration Activities requires considerably frequent communication between and among requirements developers and stakeholders, which will certainly result in unacceptably high cost under some circumstances.
2. One of the most important sources of cost is the collection and management of structured specifications in nature language. We need to find a better way to process raw data from stakeholders.
3. Another source of cost is the ubiquitous involvement of stakeholders. We should decide whether a project is adaptive in applying SoF methods at the beginning of each project. Some projects with relatively constant and determinate requirements are not supposed to apply SoF methods.

Unfinished Work:

- SoF process exactly describes each step of early phase of requirements development, but there are still some steps rely on intuition and experience of requirements developers, e.g., how to extract effective high-level goals from the first structured specification in nature language.
- Current SoF methods lack a guide to the process of acquisition, analysis and validation of highest-level goals, which requires that requirements developers should make decisions under circumstance of inadequate information. These decisions cannot be reasonably evaluated.
- Another topic omitted by SoF is how to use goals to organize scenarios, that is how to link strategic scenarios to tactical scenarios(i.e., high-level goals).

Unfinished Work:

- SoF process exactly describes each step of early phase of requirements development, but there are still some steps rely on intuition and experience of requirements developers, e.g., how to extract effective high-level goals from the first structured specification in nature language.
- Current SoF methods lack a guide to the process of acquisition, analysis and validation of highest-level goals, which requires that requirements developers should make decisions under circumstance of inadequate information. These decisions cannot be reasonably evaluated.
- Another topic omitted by SoF is how to use goals to organize scenarios, that is how to link strategic scenarios to tactical scenarios(i.e., high-level goals).

Q & A