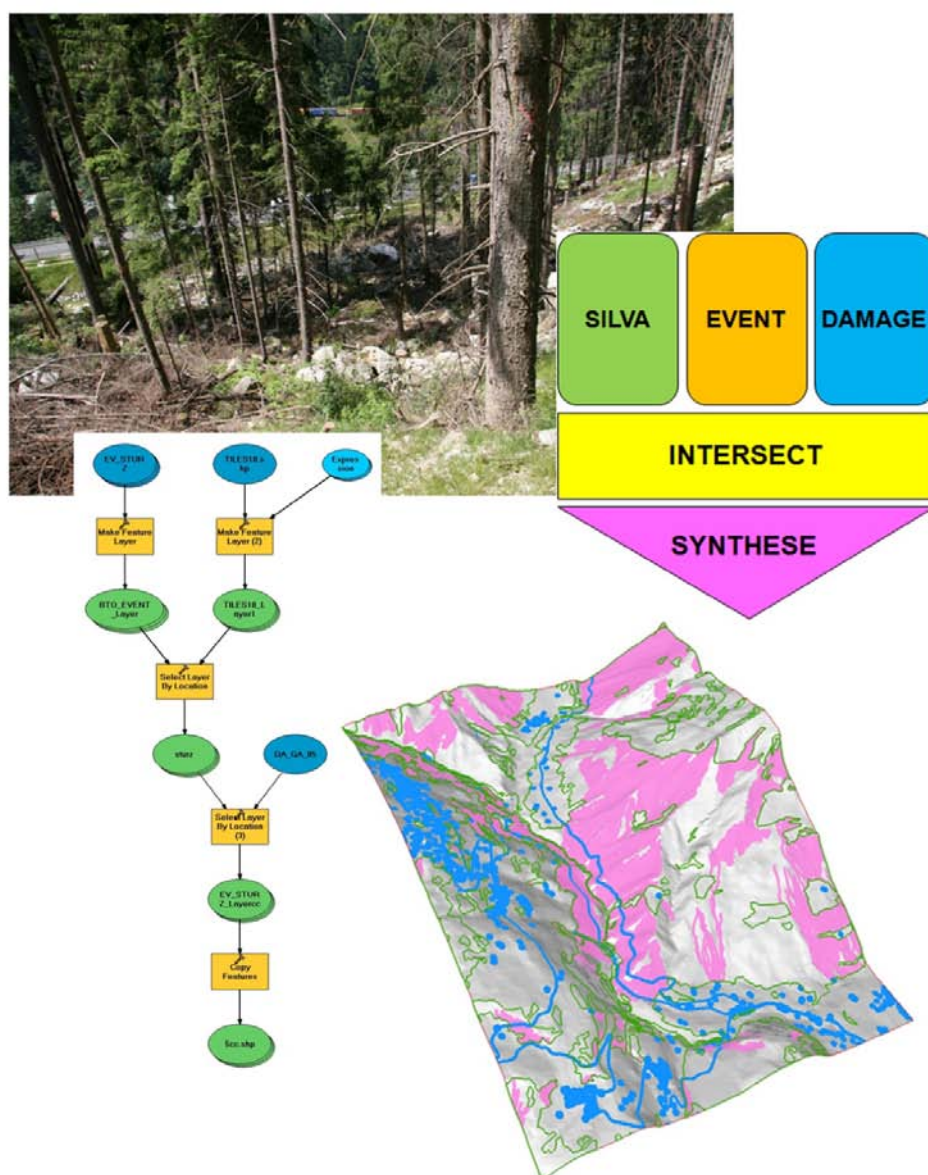




Janvier 2013 / LOS

# SilvaProtect-CH: Manuel SIG

## Annexe 3



(Photo S. Losey, 2011)



## Table des matières

|       |                                                                         |    |
|-------|-------------------------------------------------------------------------|----|
| 1     | Introduction .....                                                      | 5  |
| 2     | Généralités informatiques.....                                          | 6  |
| 2.1   | Hardware et Software .....                                              | 6  |
| 2.2   | Gestion de la taille des données .....                                  | 7  |
| 2.3   | Des trajectoires aux surfaces .....                                     | 8  |
| 2.4   | Solutions pour les modélisations longues .....                          | 9  |
| 2.5   | Utilisation des couleurs.....                                           | 13 |
| 2.6   | Données de base (BASIC) .....                                           | 13 |
| 3     | Processus de dangers (EVENT) .....                                      | 18 |
| 3.1   | Avalanches .....                                                        | 18 |
| 3.2   | Chutes de pierres .....                                                 | 19 |
| 3.3   | Coulées de boue / glissements spontanés.....                            | 20 |
| 3.4   | Processus liés aux cours d'eau .....                                    | 20 |
| 4     | Enjeux (DAMAGE).....                                                    | 23 |
| 4.1   | Le réseau ferroviaire.....                                              | 25 |
| 4.2   | Les installations .....                                                 | 27 |
| 4.3   | Les bâtiments .....                                                     | 32 |
| 4.3.1 | Maisons d'habitation habitées en permanence .....                       | 33 |
| 4.3.2 | Maisons d'habitation habitées périodiquement .....                      | 34 |
| 4.3.3 | Bâtiments industriels.....                                              | 35 |
| 4.3.4 | Bâtiments publics.....                                                  | 44 |
| 4.3.5 | Regroupement des bâtiments de pondérations 3 et 5 .....                 | 45 |
| 4.4   | Le réseau routier.....                                                  | 46 |
| 4.5   | Combinaison des enjeux pour le calcul des indices.....                  | 48 |
| 4.5.1 | Enjeux pour l'indice de forêts protectrices .....                       | 48 |
| 4.5.2 | Enjeux utilisés pour le calcul de l'indice du potentiel de dommage..... | 50 |
| 5     | Processus pertinents (INTERSECT) .....                                  | 55 |
| 5.1   | Avalanches: zones de déclenchement pertinentes .....                    | 55 |
| 5.1.1 | Description de la modélisation.....                                     | 55 |
| 5.1.2 | Enjeux selon les régions d'avalanches.....                              | 57 |
| 5.1.3 | Zones de déclenchement pertinentes .....                                | 59 |
| 5.2   | Chutes de pierres: trajectoires pertinentes.....                        | 60 |
| 5.2.1 | Description de la modélisation.....                                     | 60 |
| 5.2.2 | Trajectoires pertinentes pour un carré de 40 km.....                    | 61 |
| 5.2.3 | Trajectoires pertinentes pour toute la Suisse .....                     | 63 |
| 5.3   | Glissements de terrain: trajectoires pertinentes .....                  | 64 |
| 5.3.1 | Description de la modélisation.....                                     | 64 |
| 5.3.2 | Trajectoires pertinentes pour un carré de 40 km.....                    | 65 |
| 5.3.3 | Trajectoires pertinentes pour toute la Suisse .....                     | 67 |
| 5.4   | Processus liés aux lits pertinents .....                                | 68 |

|       |                                                                          |     |
|-------|--------------------------------------------------------------------------|-----|
| 5.4.1 | Lits pertinents à l'aide des débordements avec dépôts .....              | 69  |
| 5.4.2 | Lits pertinents à l'aide des laves torrentielles .....                   | 75  |
| 5.4.3 | Processus pour les lits pertinents .....                                 | 80  |
| 5.5   | Groupement des processus pertinents .....                                | 86  |
| 6     | Surface forestière (SILVA).....                                          | 87  |
| 7     | Indices de forêts protectrices et du potentiel de danger (SYNTHESE)..... | 89  |
| 7.1   | Indices des forêts protectrices.....                                     | 89  |
| 7.2   | Indice du potentiel de dommages.....                                     | 92  |
| 8     | Littérature.....                                                         | 95  |
| 9     | Annexes.....                                                             | 96  |
| 9.1   | Modèle de données .....                                                  | 96  |
| 9.1.1 | Données utilisées et obtenues .....                                      | 96  |
| 9.1.2 | Description des différents modèles .....                                 | 101 |
| 9.2   | Aperçu des modèles de traitement de ModelBuilder.....                    | 110 |
| 9.3   | Script Python pour les avalanches .....                                  | 114 |
| 9.3.1 | Pp_Launch.py: départ.....                                                | 114 |
| 9.3.2 | Pp_Programm.py: déclencheur des différentes étapes .....                 | 117 |
| 9.3.3 | Pp_proc_rel.py: toutes les étapes de calcul et méthode.....              | 118 |
| 9.3.4 | Pp_Settings.py: paramétrage et réglage .....                             | 132 |
| 9.3.5 | Pp_util_GIS.py: appel objets SIG et avertissements.....                  | 133 |
| 9.4   | Temps de calcul.....                                                     | 136 |



# 1 Introduction

L'objectif du manuel SIG est (1) d'expliquer de manière claire les différentes modélisations effectuées à l'aide d'un système d'informations géo-référencées dans le projet SilvaProtect-CH, (2) de pouvoir reproduire le calcul des indices de forêts protectrices (IFP) et du potentiel de danger (IPD) et (3) d'apporter une certaine méthodique pour les projets de grande envergure utilisant des données géo-référencées.

La structure du projet est décrite dans le chapitre 2.2 du rapport principal. Les liens entre les chapitres de ce manuel et les différents modules sont représentés dans la figure ci-dessous.

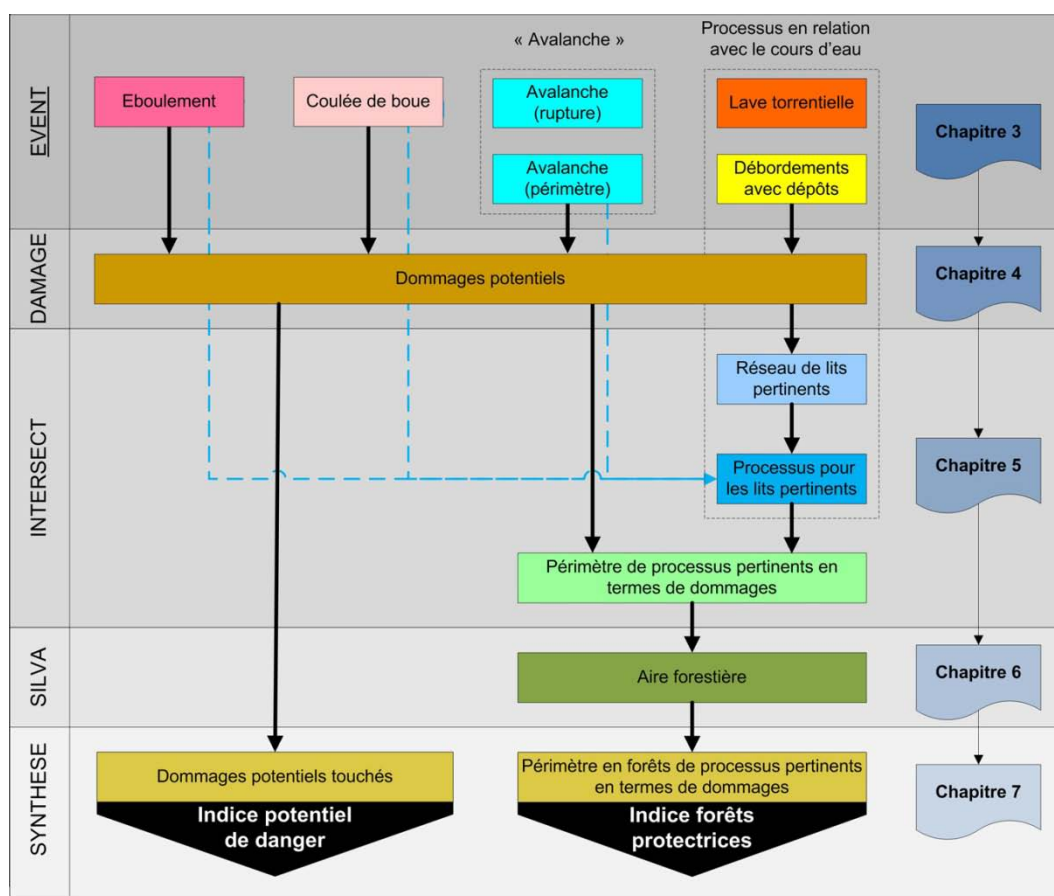


Figure 1: Liens entre les modules du projet SilvaProtect-CH et le manuel SIG.

Le chapitre 2 s'ajoute aux chapitres énoncés dans la Figure 2 et donne quelques généralités sur les programmes utilisés et sur l'organisation des données géo-référencées.

## 2 Généralités informatiques

### 2.1 Hardware et Software

Tous les calculs sont effectués sur une station locale et les données sont sauvegardées sur un disque dur externe. En effet, à cause de la taille importante des données traitées, l'utilisation d'une station en réseau ralentirait le réseau lui-même et ses utilisateurs.

Les programmes suivants ont été utilisés dans le cadre du projet SilvaProtect-CH:

- Arc GIS 9.3: Gestion des données géo-référencées
- ModelBuilder: Géotraitement pour la modélisation
- Python 2.5: Script de géotraitement pour la modélisation des avalanches

L'organisation des données géo-référencées sont présentées dans la figure ci-dessous:

- D:\Grunddaten: Carte topographique au 25'000
- D:\SilvaProtect:
  - o ADMIN: Informations administratives sur le projet
  - o BASIC: Données de base (chapitre 2.6)
  - o DAMAGE: Données et calcul du potentiel de dommage (chapitre 4)
  - o EVENT: Processus de dangers naturel (chapitre 3)
  - o INTERSECT: Processus pertinents (chapitre 5)
  - o SILVA: Surface forestière (chapitre 6)
  - o SYNTHESE: Résultats obtenus (chapitre 7)
  - o TOOLBOX: Modèles de géotraitement préparés avec ModelBuilder (Annexe 9.1)
- D:\Temp Silva: Données temporaires afin d'effectuer des étapes intermédiaires
- D:\Workspace: Données temporaires calculées dans un modèle de géotraitement et directement supprimées

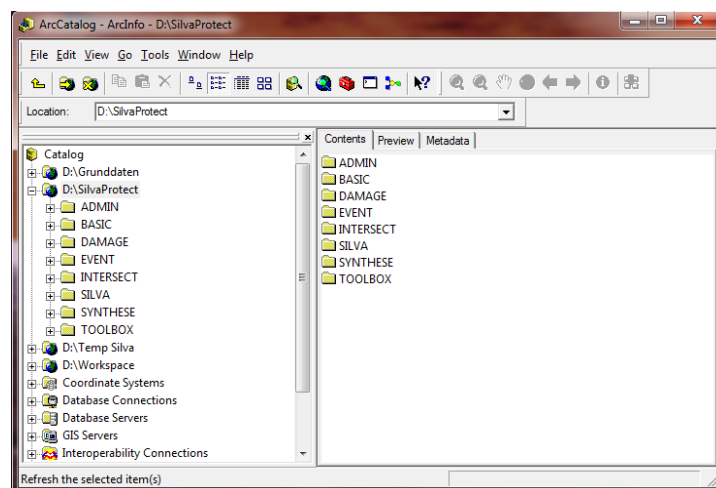


Figure 2: Organisation des données géo-référencée sur la station de travail.

## 2.2 Gestion de la taille des données

La taille des données utilisées dans le projet (par exemple: chutes de pierres: 9.3 millions de trajectoires, 1.2 GB ou glissements de terrain: 47.6 millions de trajectoires, 6.5 GB) pose un certain nombre de problèmes lors de modélisation; en effet, Arc GIS n'arrive pas toujours à gérer une telle quantité de données et se bloque souvent. Afin d'éviter ce problème, la Suisse a été divisée en carrés de différentes tailles et les données principales ont été sélectionnées selon la taille des carrés. Cette sélection respecte les règles suivantes:

- Différentes tailles des carrés: Ainsi, selon la taille de la donnée de base, on utilisera un carré d'une taille différentes.
- Emboîtement des carrés: Ainsi lorsqu'un carré est trop grand, on peut toujours le divisé en d'autres carrés.
- Numérotation précise: Ainsi les relations entre les carrés de différentes tailles sont claires et traçables.

La Suisse est donc divisée en 49 carrés ayant un côté de 40 kilomètres et numérotés du nord-ouest au sud-est (Figure 3).

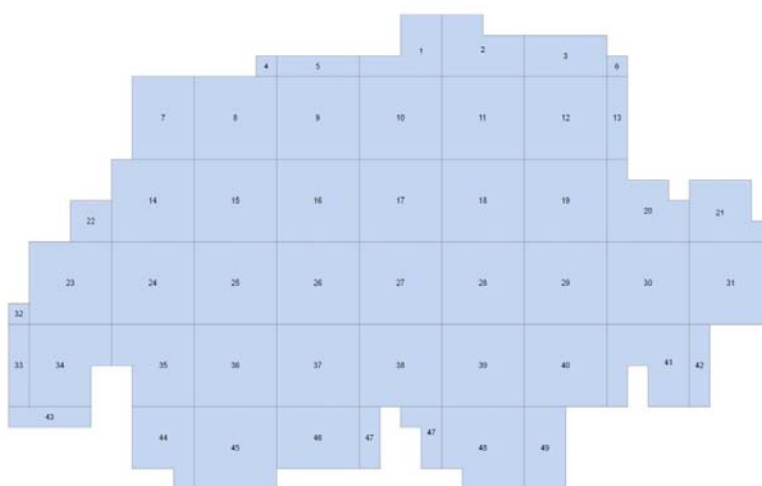


Figure 3: La Suisse divisée en 49 carrés.

Chaque carré est ensuite divisé en 4 carrés de 20 km de côté et 16 carrés de 10 km de côté.



Figure 4: Division des carrés de 40 kilomètres de coté.

Cette méthode facilite l'implémentation du géotraitement à l'intérieur de ModelBuilder.

## 2.3 Des trajectoires aux surfaces

Même si les plus petits carrés ont un côté de 10 km, il arrive souvent d'avoir plus de 200'000 trajectoires sur cette surface. La fonction "Dissolve" de Arc GIS n'arrive pas à gérer une telle quantité de données. Afin de transformer les vecteurs "lignes" en vecteurs "polygones", il est plus facile et rapide de passer par un raster.

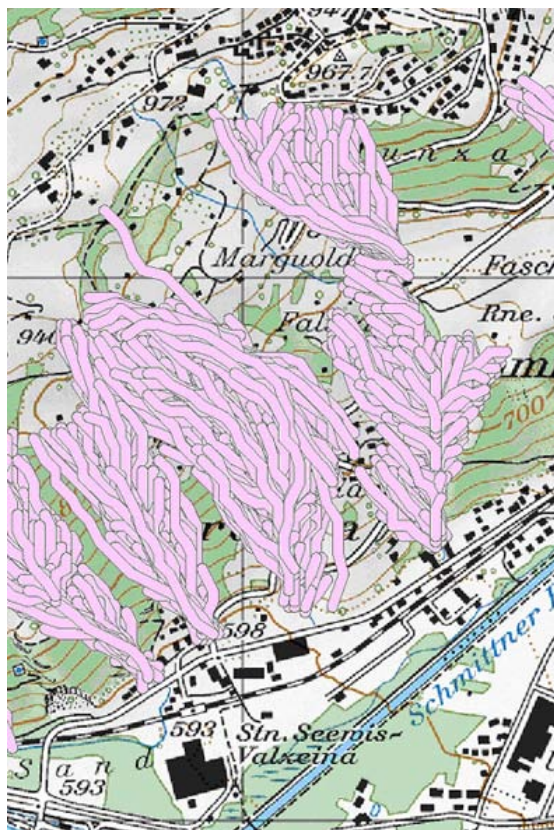


Figure 5: Trajectoires de glissements de terrain avec une zone tampon de 10 mètres.

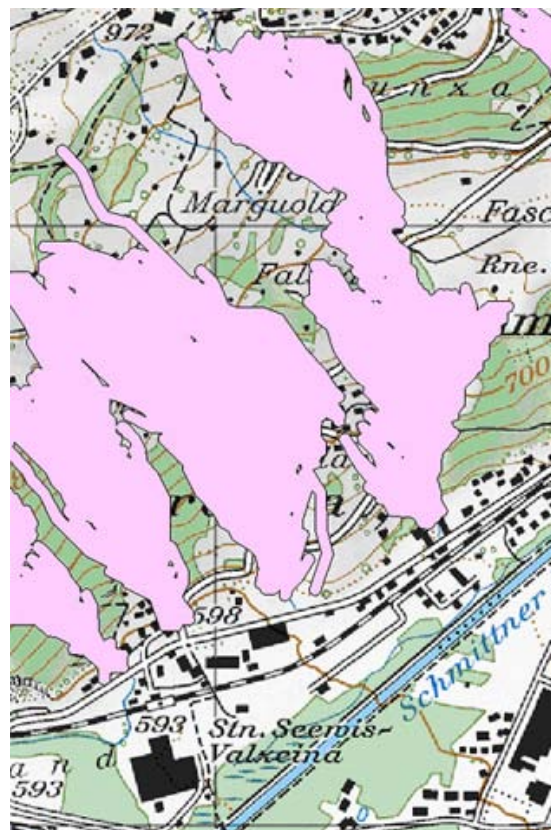
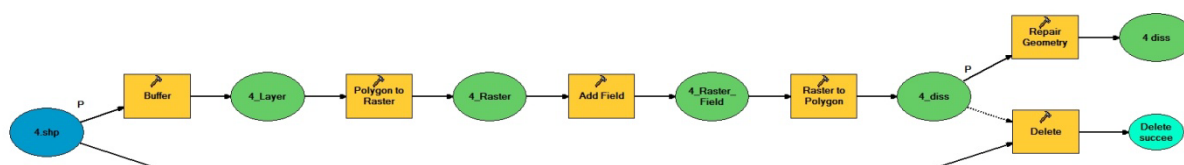


Figure 6: Trajectoires transformées en polygones.

Dans le cadre du projet SilvaProtect-CH, les trajectoires des différents processus sont toujours transformées en polygones et la méthode décrite ci-dessous est utilisée.



Modèle 1: Transformation de lignes en polygones.

**Détails du modèle:**

- **Buffer** Distance: 10 m
- **Polygon to Raster** Value Field: FID; Output Raster Dataset: "enlever le format .img; ainsi une table avec attribut sera créé"; Cellsize: 1
- **Add Field** Field Name: Diss; Field Type: short
- **Raster to Polygon** Field: Diss; Simplify Polygon checked

Les deux „P“ dans le modèle permettent de définir le choix de la donnée à introduire et le nom de la donnée qui va ainsi être produite. Ce modèle peut donc être utilisé pour tous les carrés, décrit dans le chapitre précédent, et toute la Suisse peut être ainsi traitée.

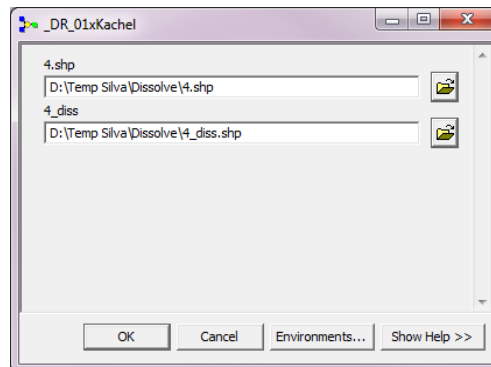


Figure 7: Paramètres du Modèle 1

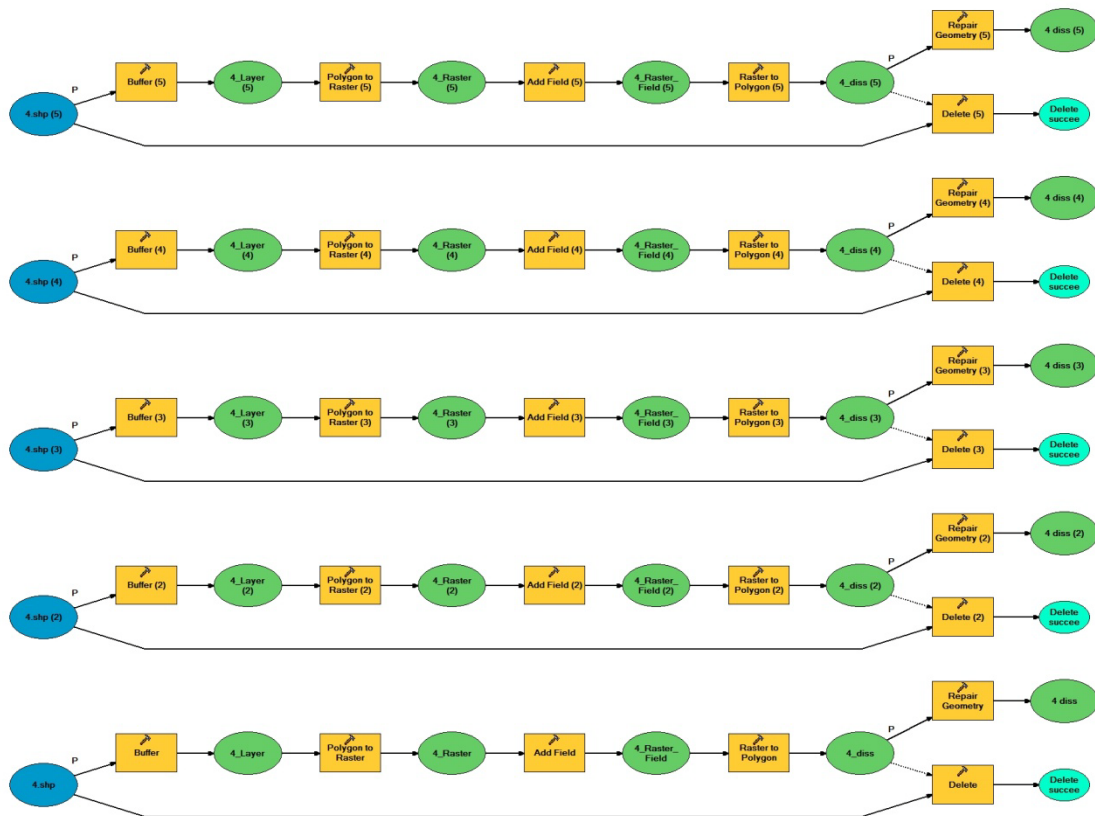
## 2.4 Solutions pour les modélisations longues

Lors de modélisations sur toute la Suisse, il est important que cela puisse se faire de manière automatique, avec le minimum d'interventions externes. ModelBuilder a été utilisé car il demande très peu de connaissance en programmation et est très visuel.

Pour les modélisation longues, trois méthodes ont été utilisées.

### Méthode 1: plusieurs processus dans un même modèle

Cette méthode (Modèle 2) est très intéressante si le calcul n'est pas dépendant de la localisation géospatiale, mais se veut très flexible. Lorsque l'on ouvre la fenêtre des paramètres (Modèle 2), on peut changer les données à introduire et les noms des données produites selon les besoins. Par contre l'utilisateur doit toujours introduire ces informations lui-même. Afin de diminuer les interventions la méthode 2 a été développée.



Modèle 2: Plusieurs processus dans le même modèle; ici le Modèle 1.

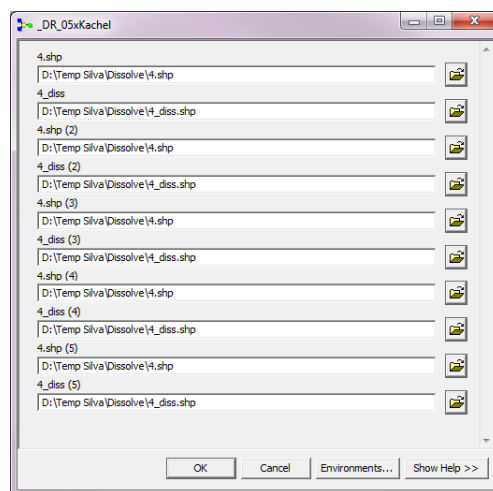


Figure 8: Paramètres du Modèle 2

## Méthode 2: modèle avec une liste de valeurs

Afin de choisir cette option, il faut sélectionner les propriétés des données à introduire à l'intérieur du modèle et choisir l'option "liste de valeurs" (Figure 9). Grâce à cela, on peut choisir plusieurs données d'entrée, sans pour autant "voir" plusieurs processus à l'intérieur du modèle (Modèle 3), ainsi plusieurs carrés peuvent être calculés en une fois (Figure 10). Par contre, les résultats intermédiaires sauvegardés doivent avoir un nom différent (Figure 11).



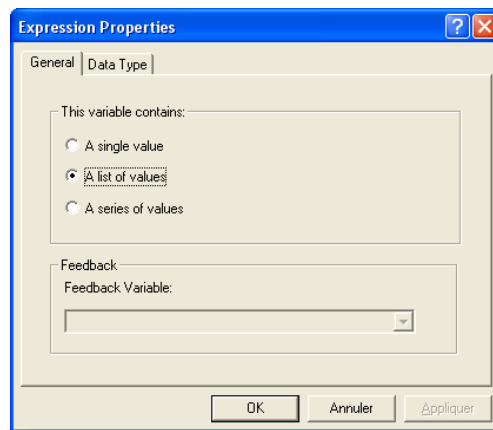
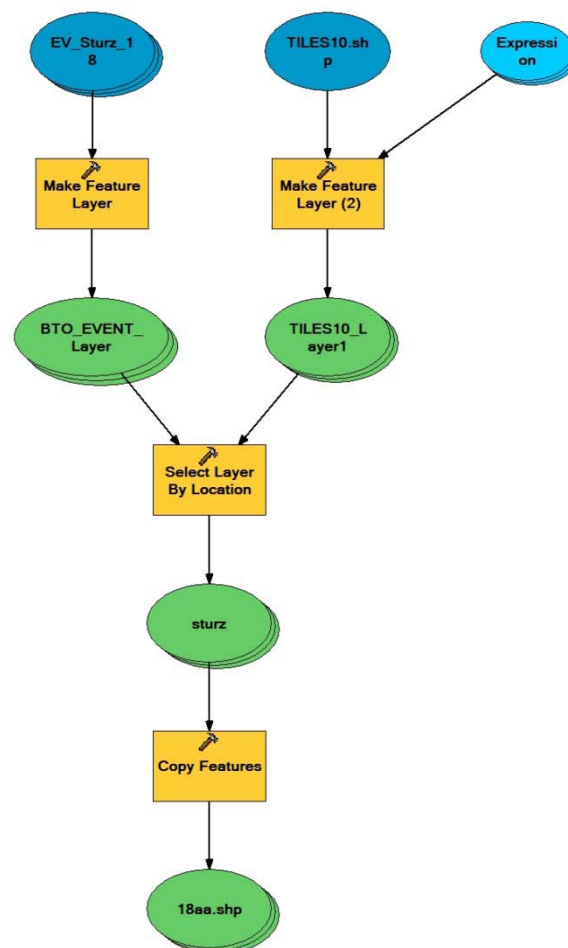


Figure 9: Propriété d'un modèle afin d'obtenir une liste de valeurs

Quand un modèle est construit de cette manière, alors toutes les données vectorielles introduites doivent avoir la propriété "liste de valeurs".



Modèle 3: Exemple d'un modèle avec une liste de valeurs.

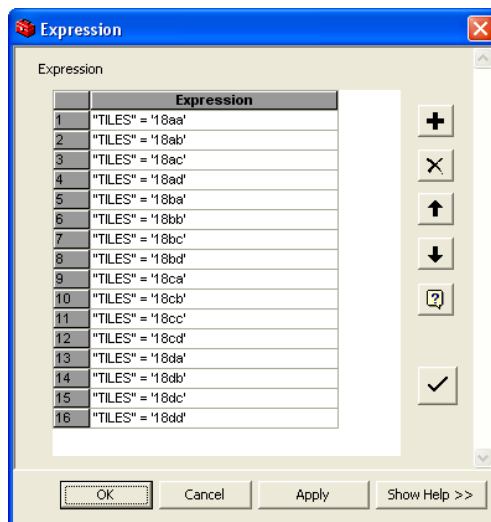


Figure 10: Propriété de l'expression du Modèle 3.

Avec ce type de modèle et la division de la Suisse en carrés numérotés de manière systématique, un nouveau modèle pour un autre carré peut être facilement élaboré. Dans notre cas, on peut sans autre remplacer le carré n° 18 par le carré n° 19 et ainsi de suite.

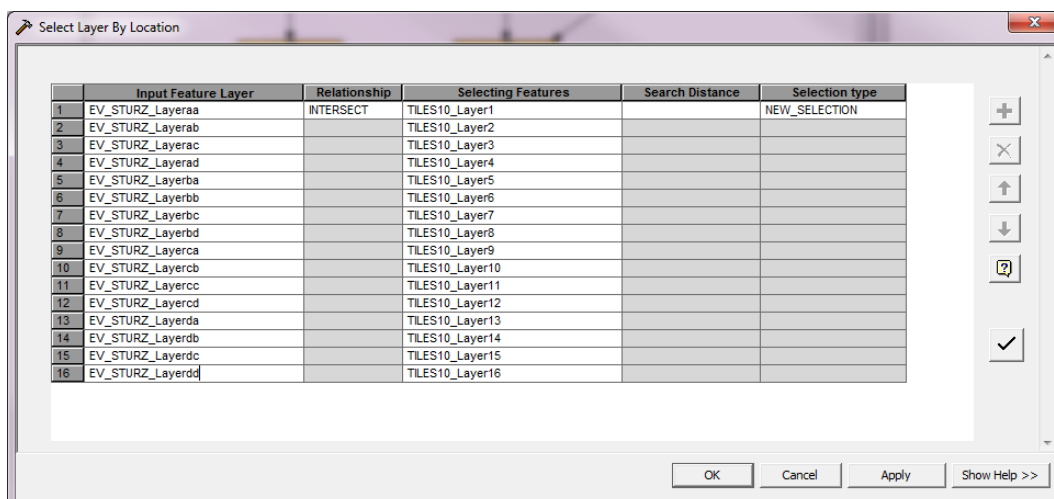


Figure 11: Propriété de "Select by location du Modèle 3

Pour les résultats intermédiaires qui ne seront pas sauvegardés, il est important qu'à l'intérieur du modèle il y ai des noms différents pour les résultats générés; par contre si le modèle est utilisé plusieurs fois de suite, il n'est pas nécessaire de renommer ces résultats, puisqu'on ne va pas les conserver.

### Méthode 3: plusieurs modèles implémentés dans un modèle principal

Cette solution permet de calculer plusieurs modèles précédemment créés pour autant que les données introduites soient clairement définies, comme c'est le cas dans la méthode 2. Ainsi toute la Suisse peut être modélisée d'un seul trait en prenant carré après carré selon la Figure 3 et la Figure 4. Le



modèle suivant permet de définir les trajectoires de glissements de terrain entrant en contact avec un enjeu; selon ce dernier, le temps de calcul pour l'ordinateur peut durer plus de 50 heures.



Modèle 4: Modèles implémentés dans un modèle principal.

## 2.5 Utilisation des couleurs

Plusieurs couleurs ont été utilisées dans le cadre de ce manuel, en voici leur signification:

- **Bleu** nom d'une donnée vectorielle introduite dans un modèle
- **Vert** nom d'une donnée vectorielle générée par un modèle
- **Rouge** nom d'un modèle élaboré à l'aide de ModelBuilder

## 2.6 Données de base (BASIC)

Ces données vectorielles sont utilisées pour effectuer les différentes modélisations. Il existe trois types de groupes:

- Les données politiques: Suisse et cantons,
- Les données selon la taille de calcul: carrés de 10, 20 et 40 kilomètres et
- Les données selon le thème: avalanches et débordements avec dépôts.

La Suisse ([CH.shp](#)):

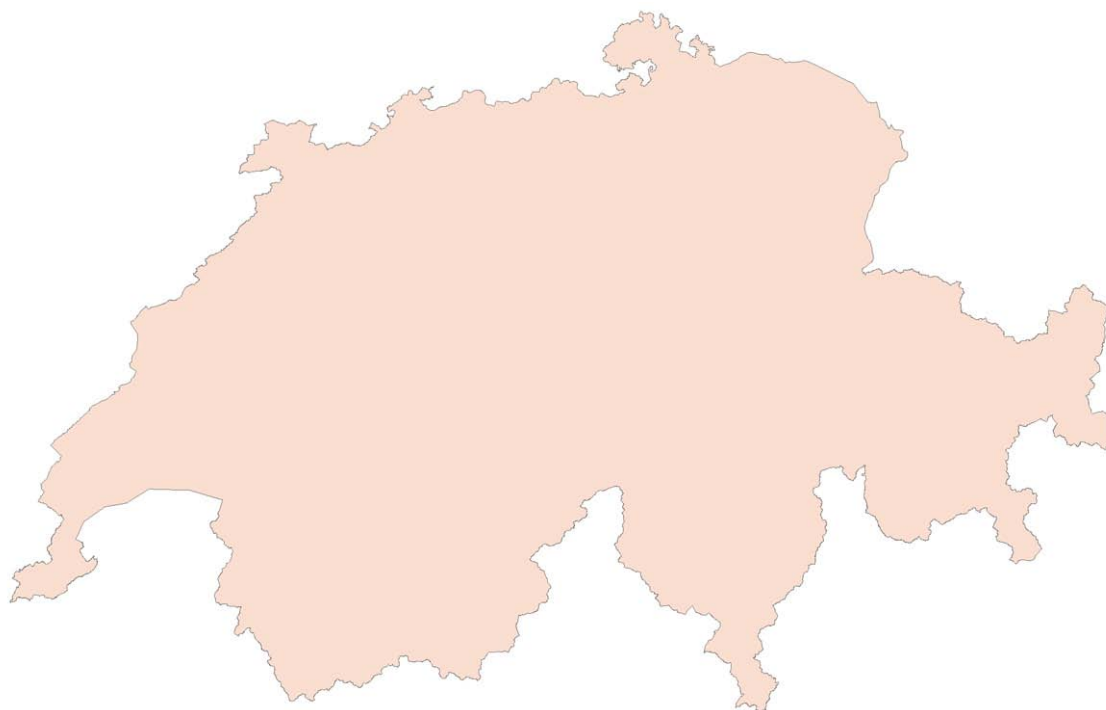


Figure 12: Donnée vectorielle pour la Suisse.

Les cantons: ([Kantone.shp](#)):

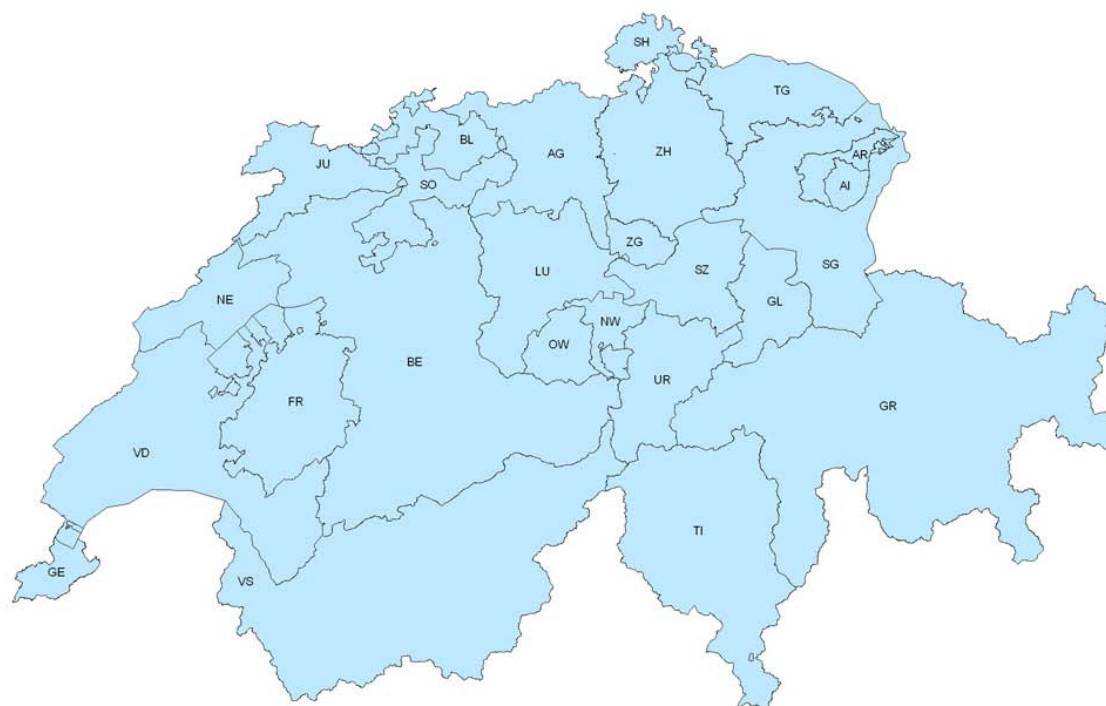


Figure 13: Donnée vectorielle pour les cantons.

Carrés de 10 km de côtés divisant la Suisse ([TILES10.shp](#)):



Figure 14: Donnée vectorielle pour les carrés de 10 kilomètres.

Carrés de 20 km de côtés divisant la Suisse ([TILES20.shp](#)):

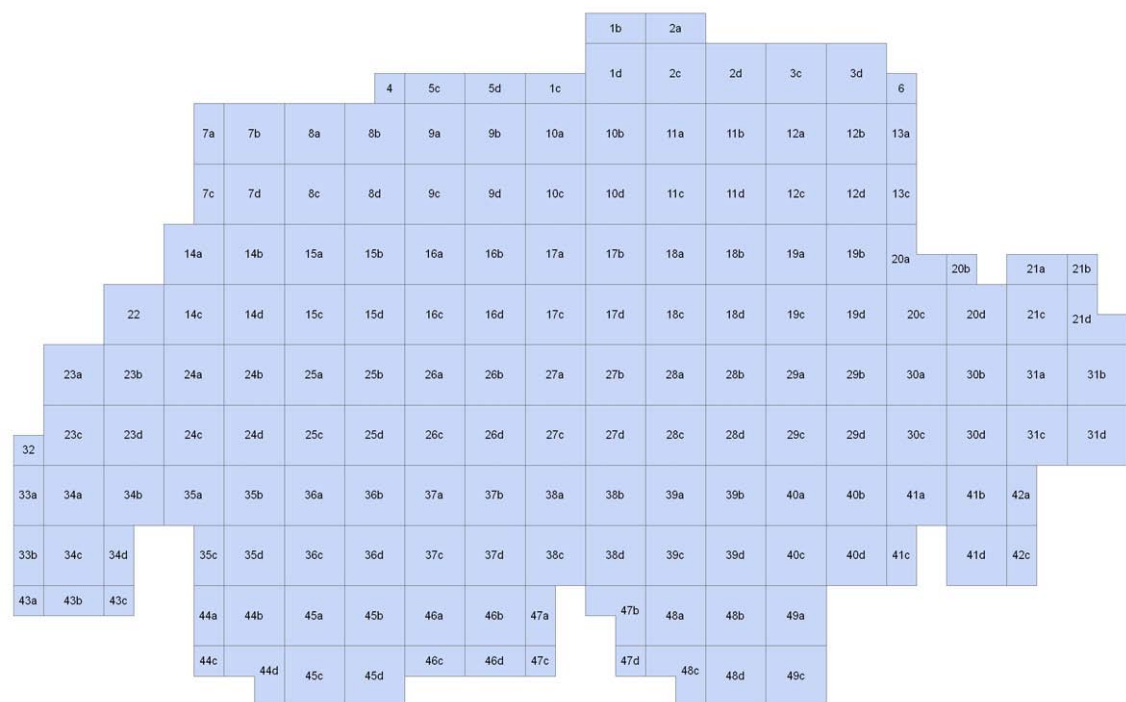


Figure 15: Donnée vectorielle pour les carrés de 20 kilomètres.

Carrés de 40 km de côtés divisant la Suisse ([TILES40.shp](#)):

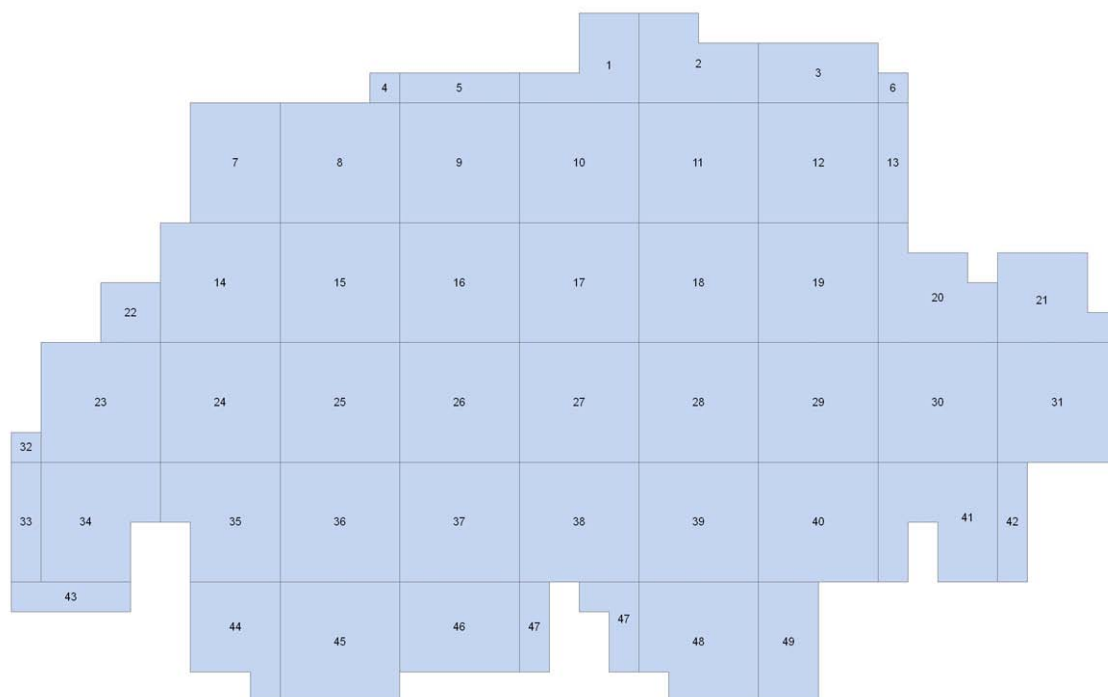


Figure 16: Donnée vectorielle pour les carrés de 40 kilomètres.

Régions d'avalanches pour la modélisation des zones de déclenchement ([BA\\_Lawine\\_Gebiete.shp](#)):

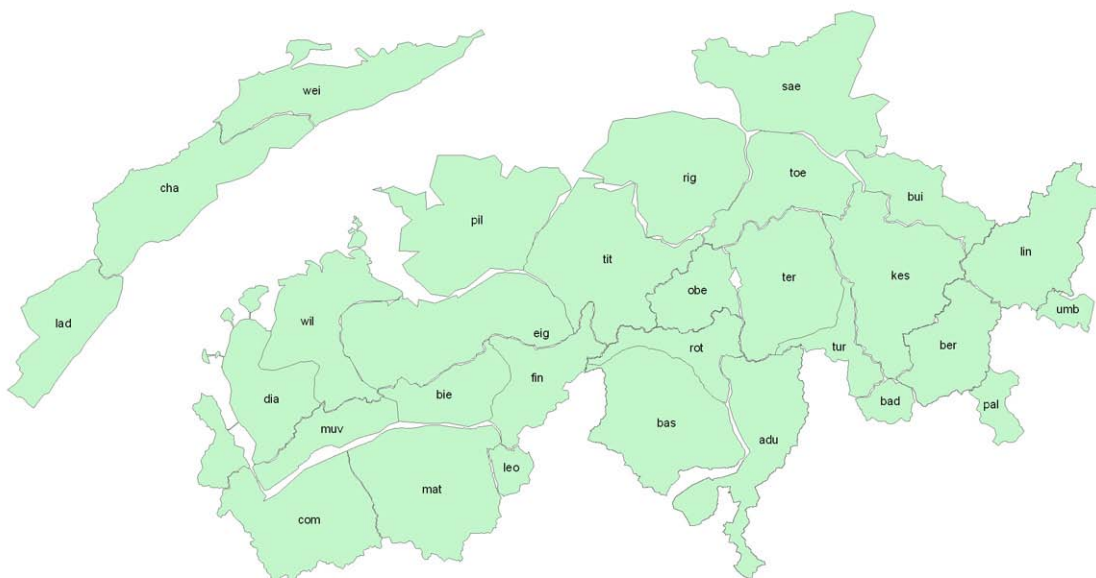


Figure 17: Donnée vectorielle pour les régions d'avalanches.

Régions de débordements avec dépôts pour la modélisation des débordements pertinents  
([Übersarungsgebiete.shp](#)):



Figure 18: Donnée vectorielle pour les régions de débordements avec dépôts.



### 3 Processus de dangers (EVENT)

Les processus suivants ont été considérés et modélisés dans le cadre du projet SilvaProtect-CH:

- avalanches
- chutes ou éboulements
- coulées de boue / glissements spontanés
- processus en relation avec le lit des cours d'eau

#### 3.1 Avalanches

Les avalanches sont composées d'une zone de déclenchement (départ de l'avalanche où la forêt remplit son rôle protecteur) et d'un périmètre (étendue de l'avalanche lorsqu'elle a été déclenchée).

La Suisse a été divisée en 30 régions (Figure 17). Par région (xxx), les données suivantes ont été utilisées:

- xxxn\_s\_ph\_r: périmètres des petites avalanches
- xxxn\_s\_rel\_r: zones de déclenchement des petites avalanches
- xxxn\_m\_ph\_r: périmètres des avalanches moyennes
- xxxn\_m\_rel\_r: zones de déclenchement des avalanches moyennes
- xxxn\_l\_ph\_r: périmètres des grosses avalanches
- xxxn\_l\_rel\_r: zones de déclenchement des grosses avalanches
- xxxn\_l\_cap / point: force de pression pour la modélisation des grosses avalanches, répartie sous forme de point

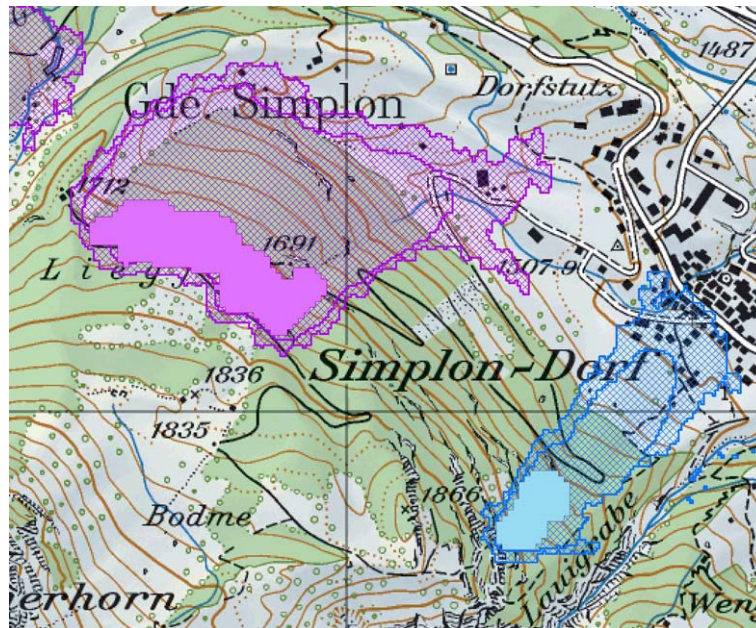


Figure 19: Périmètre d'une petite avalanche (bleu hachuré) avec sa zone de déclenchement (bleu) et d'une avalanche moyenne (violet hachuré) avec sa zone de déclenchement (violet).

| Abréviation | Nom de la région | Abréviation | Nom de la région  |
|-------------|------------------|-------------|-------------------|
| adu         | Piz Adula        | mat         | Matterhorn        |
| bad         | Piz Badile       | muv         | Muverain          |
| bas         | Basodino         | obe         | Oberalpstock      |
| ber         | Piz Bernina      | pal         | Piz Palù          |
| bie         | Bietschhorn      | pil         | Pilatus           |
| bui         | Piz Buin         | rig         | Rigi              |
| cha         | Chasseral        | rot         | Pizzo Rotondo     |
| com         | Grand Combin     | sae         | Säntis            |
| dia         | Les Diablerets   | ter         | Piz Terri         |
| eig         | Eiger            | tit         | Titlis            |
| fin         | Finsteraarhorn   | toe         | Tödi              |
| kes         | Piz Kesch        | tur         | Piz Turba         |
| lad         | La Dôle          | umb         | Pizzo Umbreil     |
| leo         | Monte Leone      | wie         | Weissenstein (JU) |
| lin         | Piz Linard       | wil         | Wilerhorn         |

Tableau 1: Abréviation et nom des régions pour la modélisation des avalanches.

### 3.2 Chutes de pierres

Les forêts permettent de réduire la vitesse cinétique des pierres et des blocs. La modélisation des trajectoires des chutes de pierres a été définie dans l'annexe 1 et le résultat est de 9.3 millions de trajectoires ([EV\\_Sturz](#)). Selon les explications données dans le chapitre 2.2, cette donnée vectorielle a été divisée en 49 carrés de 40 km (Figure 3): [EV\\_Sturz\\_01](#) à [EV\\_Sturz\\_49](#)).

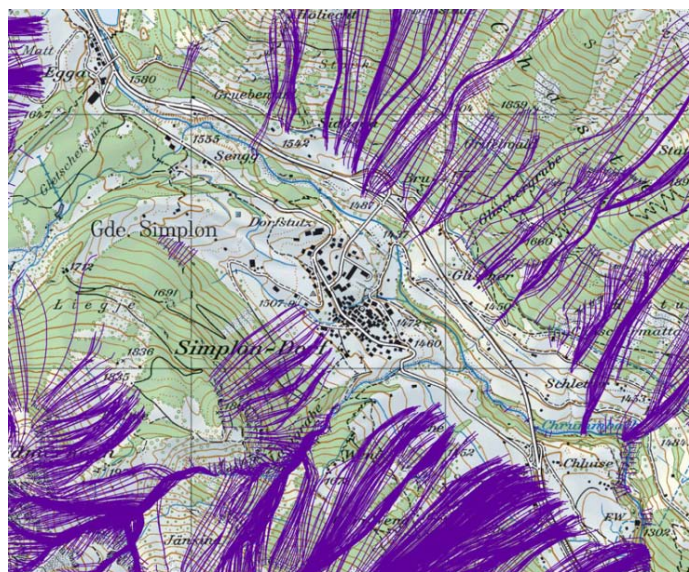


Figure 20: Trajectoires des chutes de pierres (violet).



### 3.3 Coulées de boue / glissements spontanés

Grâce à leurs racines, les forêts empêchent les coulées de boue et les glissements spontanés de se produire. La modélisation des trajectoires des coulées de boue a été définie dans l'annexe 1 et le résultat est de 47.6 millions de trajectoires ([EV\\_Hangmure](#)). Selon les explications données dans le chapitre 2.2, cette donnée vectorielle a été divisée en 49 carrés de 40 km (Figure 3): [EV\\_Hangmure\\_01](#) à [EV\\_Hangmure\\_49](#)).

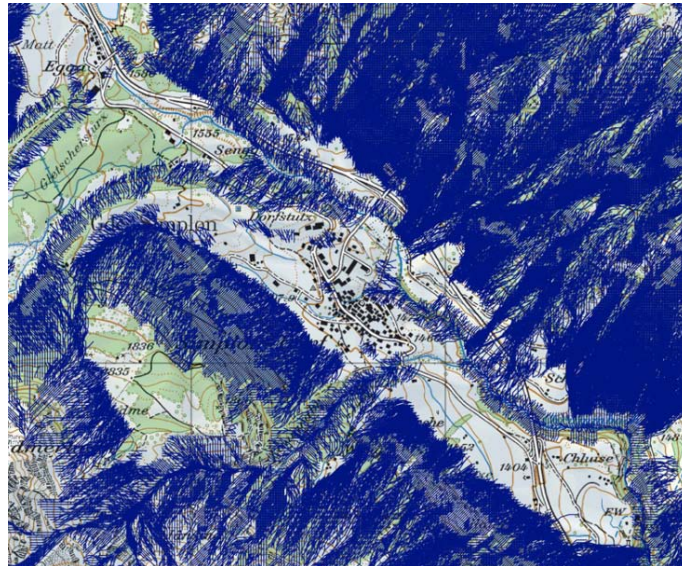


Figure 21: Trajectoires des coulées de boue (bleu).

### 3.4 Processus liés aux cours d'eau

Les forêts empêchent que les processus décrits précédemment apportent du matériel dans les cours d'eau pertinents en termes de dommages. Afin de déterminer ces cours d'eau pertinents, les données géo-référencées des débordements avec dépôts et des laves torrentielles ont été utilisées.

#### Débordements avec dépôts

Lorsque ce matériel sort d'un ruisseau, il y a toujours un point de départ et une zone d'épandage. Ces deux aspects sont représentés par deux données géo-référencées et sont divisées selon différentes régions (Figure 18):

- [EV\\_Uebersarung\\_xxx](#): zone d'épandage
- [EV\\_Uebersarung\\_start\\_xxx](#): point de départ de la zone d'épandage

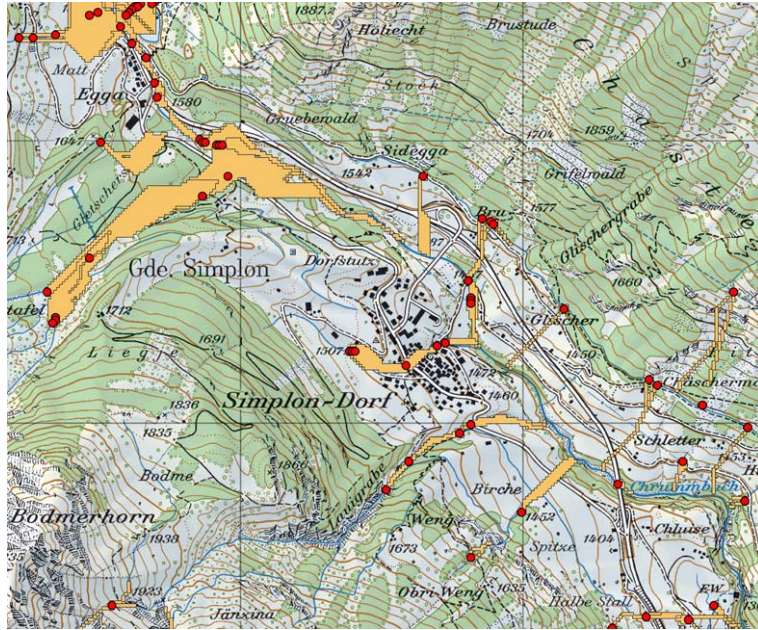


Figure 22: Zones d'épandage (beige) et leur point de départ (rouge).

| Abréviation | Nom de la région  |
|-------------|-------------------|
| aa          | Aare Alpes        |
| in          | Inn               |
| ju          | Jura              |
| mn          | Plateau Nord      |
| ms          | Plateau Sud       |
| ra          | Rhône Alpes       |
| rna         | Rhin Alpes        |
| rla         | Reuss Linth Alpes |
| rm          | Rhône Plateau     |
| rnm         | Rhin Moyen        |
| ti          | Tessin            |

Tableau 2: Abréviation et nom des régions pour la modélisation des débordements avec dépôts.

## Laves torrentielles

La modélisation des trajectoires des laves torrentielles a été définie dans l'annexe 1 et le résultat est de 6.7 millions de trajectoires ([EV\\_Murgang](#)). Selon les explications données dans le chapitre 2.2, cette donnée vectorielle a été divisée en 49 carrés de 40 km (Figure 3): [EV\\_Murgang\\_01](#) à [EV\\_Murgang\\_49](#)).

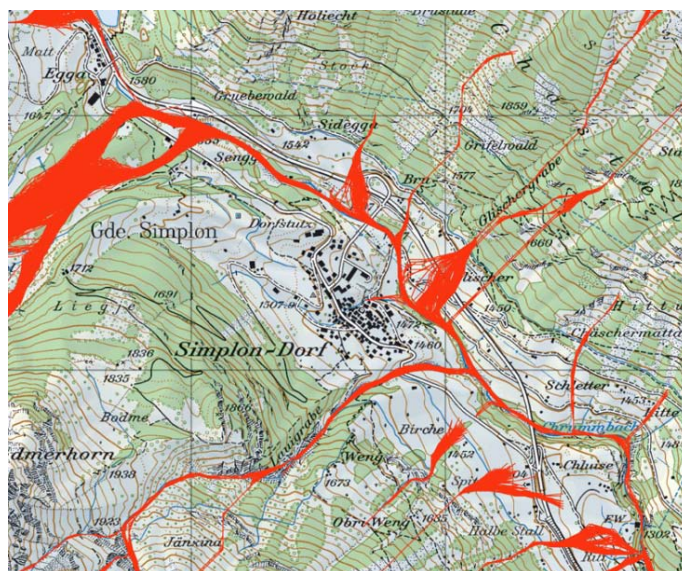


Figure 23: Trajectoires des laves torrentielles (rouge).

## 4 Enjeux (DAMAGE)

Comme le mentionne la définition de la forêt protectrice, une forêt est considérée comme telle si elle protège un enjeu reconnu contre un danger naturel ou si elle réduit les risques que ce dernier implique. Dans le chapitre 3, les dangers naturels reconnus ont été définis et géo-référencés. Dans ce chapitre, il va en être de même pour les enjeux.

Les enjeux sont répartis en groupes différents:

- Le réseau ferroviaire
- Les installations
- Les bâtiments
- Le réseau routier

**Pour le calcul de l'indice de forêts protectrices**, tous les groupes d'enjeux sont considérés de manière identique et ne sont pas pondérés.

A l'inverse, **pour le calcul de l'indice du potentiel de dommages**, les potentiels de dommages ont été répartis avec une pondération différente selon leur importance:

| Répartition | Description                                                   | Pondération |
|-------------|---------------------------------------------------------------|-------------|
| Groupe 1    | Présence humaine durable ou partiellement durable             | 10          |
| Groupe 2    | Station d'arrêt avec un haut degré de probabilité             | 5           |
| Groupe 3    | Station d'arrêt avec un faible degré de probabilité           | 3           |
| Groupe 4    | Difficultés d'approvisionnement, dommages matériel importants | 1           |
| Groupe 5    | Objets restants                                               | 0           |

Tableau 3: Répartition du potentiel de dommages selon la pondération.

Lors de superposition, seul l'enjeu avec la plus haute pondération sera retenu pour le calcul de l'indice du potentiel de dommage.

L'**abréviation swi**, située à la fin de la donnée vectorielle, signifie que cette dernière sera utilisée pour le calcul de l'indice de forêts protectrices.

Les données vectorielles suivantes sont utilisées pour la modélisation des différents potentiels de dommages et seront actualisées selon les besoins. Elles sont sauvegardées sur la station de travail sous **D:\SilvaProtect\DAMAGE\INPUT**.

| Nom de la donnée      | Année | Source                                 |
|-----------------------|-------|----------------------------------------|
| bn97hn3               | 1997  | OFS                                    |
| DA_Betriebszaehlung01 | 2008  | OFS                                    |
| DA_Volkszaehlung00    | 2000  | OFS                                    |
| GSM_Sendeanlagen      | 2008  | OFEV, section rayonnement non ionisant |
| str_25_arc            | 2008  | Swisstopo                              |
| V25_anl_polygon       | 2008  | Swisstopo                              |
| V25_eis_arc           | 2008  | Swisstopo                              |
| V25_eob_arc           | 2008  | Swisstopo                              |
| V25_eob_point         | 2008  | Swisstopo                              |
| V25_geb_polygon       | 2008  | Swisstopo                              |
| V25_gwn_arc           | 2008  | Swisstopo                              |
| V25_pri_polygon       | 2008  | Swisstopo                              |

Tableau 4: Données vectorielles utilisées pour la modélisation du potentiel de dommages.

Les modèles de géotraitement nécessaires pour cette modélisation sont sauvegardés dans la **Toolbox SiPro 93 DAMAGE** sous **D:\SilvaProtect\TOOLBOX**

## 4.1 Le réseau ferroviaire

**Modèle:** Eisenbahn GA (Toolbox: SiPro 93 DAMAGE)

**Input:** V25\_eis\_arc.

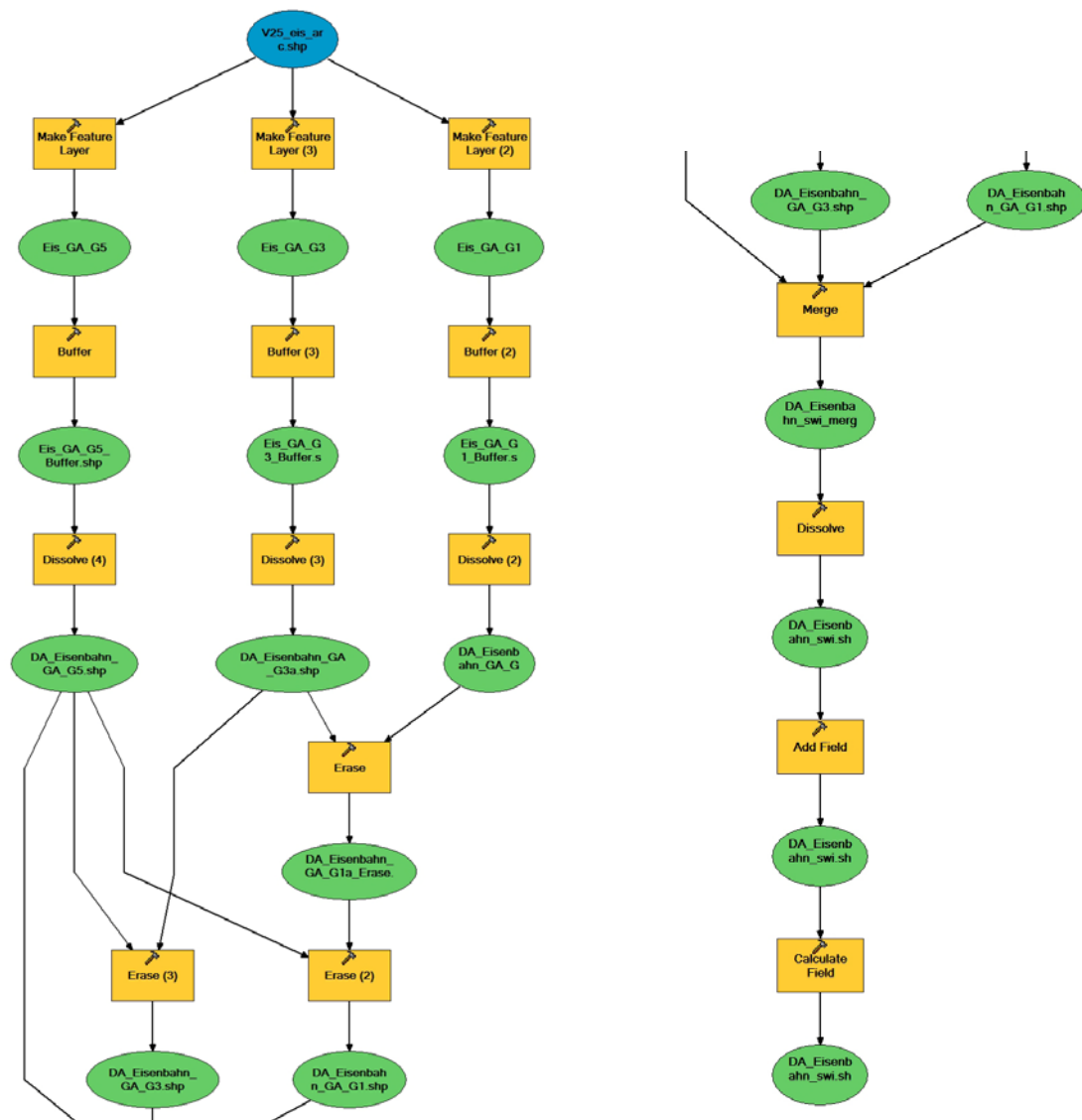
**Output:** DA\_Eisenbahn\_GA\_G1, DA\_Eisenbahn\_GA\_G3, DA\_Eisenbahn\_GA\_G5 et DA\_Eisenbahn\_swi.

### Description du modèle:

Ce modèle est subdivisé selon trois types de pondération:

- **Pondération 1:** Difficultés d'approvisionnement, dommages matériel importants
  - o Objets: voies industrielles
  - o Remarque: pas de tunnel
  
- **Pondération 3:** Station d'arrêt avec un faible degré de probabilité
  - o Objets: lignes avec desserte ferroviaire marchandise, chemins de fer à voie normale unique et à voie étroite unique
  - o Remarque: pas de tunnel
  
- **Pondération 5:** Station d'arrêt avec un haut degré de probabilité
  - o Objets: chemins de fer à voie normale multiple et à voie étroite multiple
  - o Remarque: pas de tunnel





Modèle 5: Modèle pour le réseau ferroviaire.

#### Détails du modèle:

##### GA\_G1

- **V25\_eis\_arc: Make Feature Layer:** "OBJEKTVAL" = 'I\_Geleis' AND "TUNNELTYPE" = ''

##### GA\_G3

- **V25\_eis\_arc: Make Feature Layer:** ("OBJEKTVAL" = 'Gt\_Bahn' OR "OBJEKTVAL" = NS\_BAHN1 OR "OBJEKTVAL" = 'SS\_BAHN1') AND "TUNNELTYPE" = ''

##### GA\_G5

- **V25\_eis\_arc: Make Feature Layer:** ("OBJEKTVAL" = 'NS\_BAHN2' OR "OBJEKTVAL" = 'SS\_BAHN2') AND TUNNELTYPE = ''
- **Buffer:** Distance: 10m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Erase:** Input: DA\_Eisenbahn\_GA\_G1a, Erase: DA\_Eisenbahn\_GA\_G3a
- **Erase (2):** Input: DA\_Eisenbahn\_GA\_G1a\_Erase, Erase: DA\_Eisenbahn\_GA\_G5
- **Erase (3):** Input: DA\_Eisenbahn\_GA\_G3a, Erase: DA\_Eisenbahn\_GA\_G5
- **Merge:** DA\_Eisenbahn\_GA\_G5, DA\_Eisenbahn\_GA\_G3 et DA\_Eisenbahn\_GA\_G1
- **Dissolve:** Create Multipart Feature: non checked
- **Add Field:** Field Name: DA\_Eis, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_Eis, Expression: 1

## 4.2 Les installations

**Modèle:** Anlagen GA (Toolbox: SiPro 93 DAMAGE)

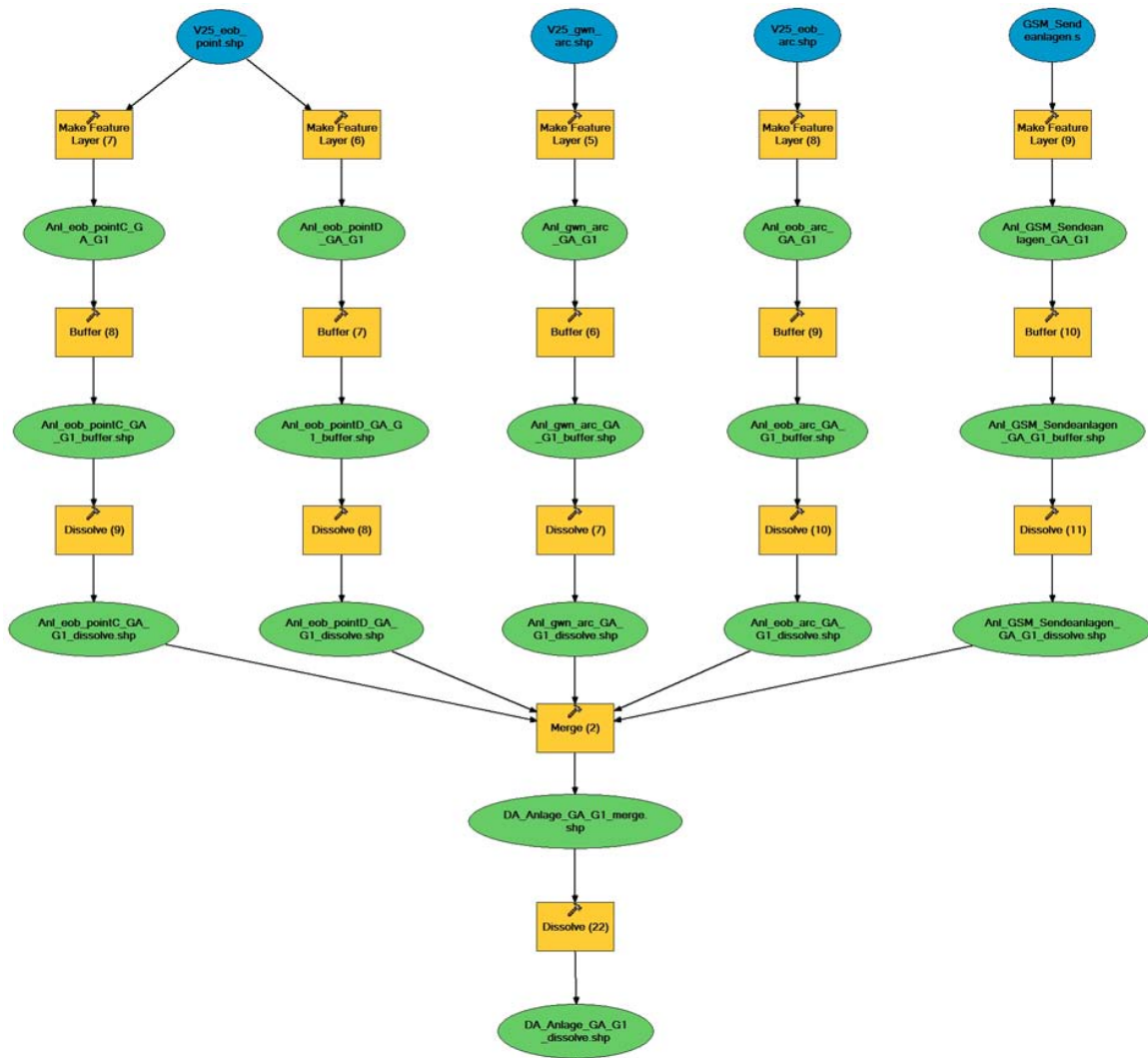
**Input:** V25\_gwn\_arc, V25\_eob\_point, V25\_eob\_arc, GSM\_Sendeanlagen, V25\_pri\_polygon, V25\_geb\_polygon et V25\_anl\_polygon

**Output:** DA\_Anlage\_GA\_G1, DA\_Anlage\_GA\_G3, DA\_Anlage\_GA\_G10 et DA\_Anlage\_swi.

### Description du modèle:

Ce modèle est subdivisé selon trois type de pondération:

- **Pondération 1:** Difficultés d'approvisionnement, dommages matériel importants
  - o Objets: cheminée marquante, réservoir (eau), château d'eau, antenne, poste émetteur, conduite forcée simple, conduite forcée multiple, station radio et poste émetteur GSM
- **Pondération 3:** Station d'arrêt avec un faible degré de probabilité
  - o Objets: digue de retenue, barrage, tour de refroidissement, réservoirs (hydrocarbures, gaz), bassin (station d'épuration des eaux, piscine), port, débarcadère, station d'épuration des eaux et centrale électrique
- **Pondération 10:** Présence humaine durable ou partiellement durable
  - o Objets: aire de gare, aire d'aéroport et aire de gare d'aéroport

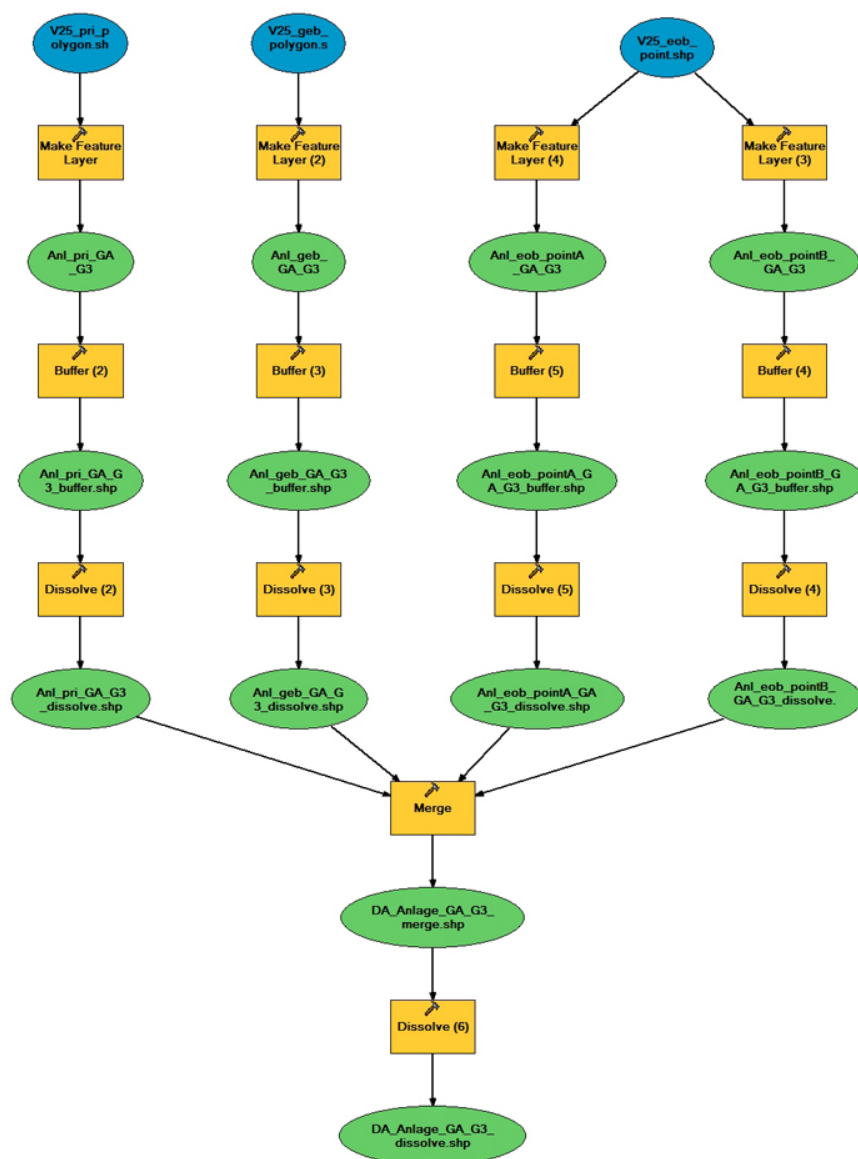


Modèle 6: Modèle pour les installations (partie 1).

#### Détails du modèle:

##### GA G1:

- **V25\_eob\_point: Make Feature Layer:** "OBJEKTVAL" = 'Kamin' OR "OBJEKTVAL" = 'Reserv' OR "OBJEKTVAL" = 'W\_Turm'
- **Buffer:** Distance: 10m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_eob\_point: Make Feature Layer:** "OBJEKTVAL" = 'Antenne' OR "OBJEKTVAL" = 'SendeAnl'
- **Buffer:** Distance: 5m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_gwn\_arc: Make Feature Layer:** "OBJEKTVAL" = 'Druckl\_1' OR "OBJEKTVAL" = 'Druckl\_2'
- **Buffer:** Distance: 5m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_eob\_arc: Make Feature Layer:** "OBJEKTVAL" = 'Sender'
- **Buffer:** Distance: 10m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **GSM\_Sendeanlagen: Make Feature Layer**
- **Buffer:** Distance: 5m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Merge:** Anl\_eob\_pointC\_GA\_G1\_dissolve, Anl\_eob\_pointD\_GA\_G1\_dissolve, Anl\_gwn\_arc\_GA\_G1\_dissolve, Anl\_eob\_arc\_GA\_G1\_dissolve et Anl\_GSM\_Sendeanlagen\_GA\_G1\_dissolve
- **Dissolve:** Create Multipart Feature: non checked

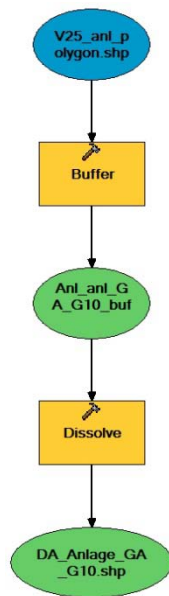


Modèle 7: Modèle pour les installations (partie 2).

### Détails du modèle

#### GA G3:

- **V25\_pri\_polygon: Make Feature Layer:** "OBJEKTVAL" = 'Z\_StauDa' OR "OBJEKTVAL" = 'Z\_StauMa'
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_geb\_polygon: Make Feature Layer:** "OBJEKTVAL" = 'Z\_Kuehlturm' OR "OBJEKTVAL" = 'Z\_Lagertank' OR "OBJEKTVAL" = 'Z\_WBecken'
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_eob\_pointA: Make Feature Layer:** "OBJEKTVAL" = 'Hafen' OR "OBJEKTVAL" = 'Schiffst'
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **V25\_eob\_pointB: Make Feature Layer:** "OBJEKTVAL" = 'ARA' OR "OBJEKTVAL" = 'EIWerk'
- **Buffer:** Distance: 10m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Merge:** Anl\_eob\_pointB\_GA\_G3\_dissolve, Anl\_eob\_pointA\_GA\_G3\_dissolve, Anl\_pri\_GA\_G3\_dissolve et Anl\_geb\_GA\_G3\_dissolve
- **Dissolve:** Create Multipart Feature: non checked

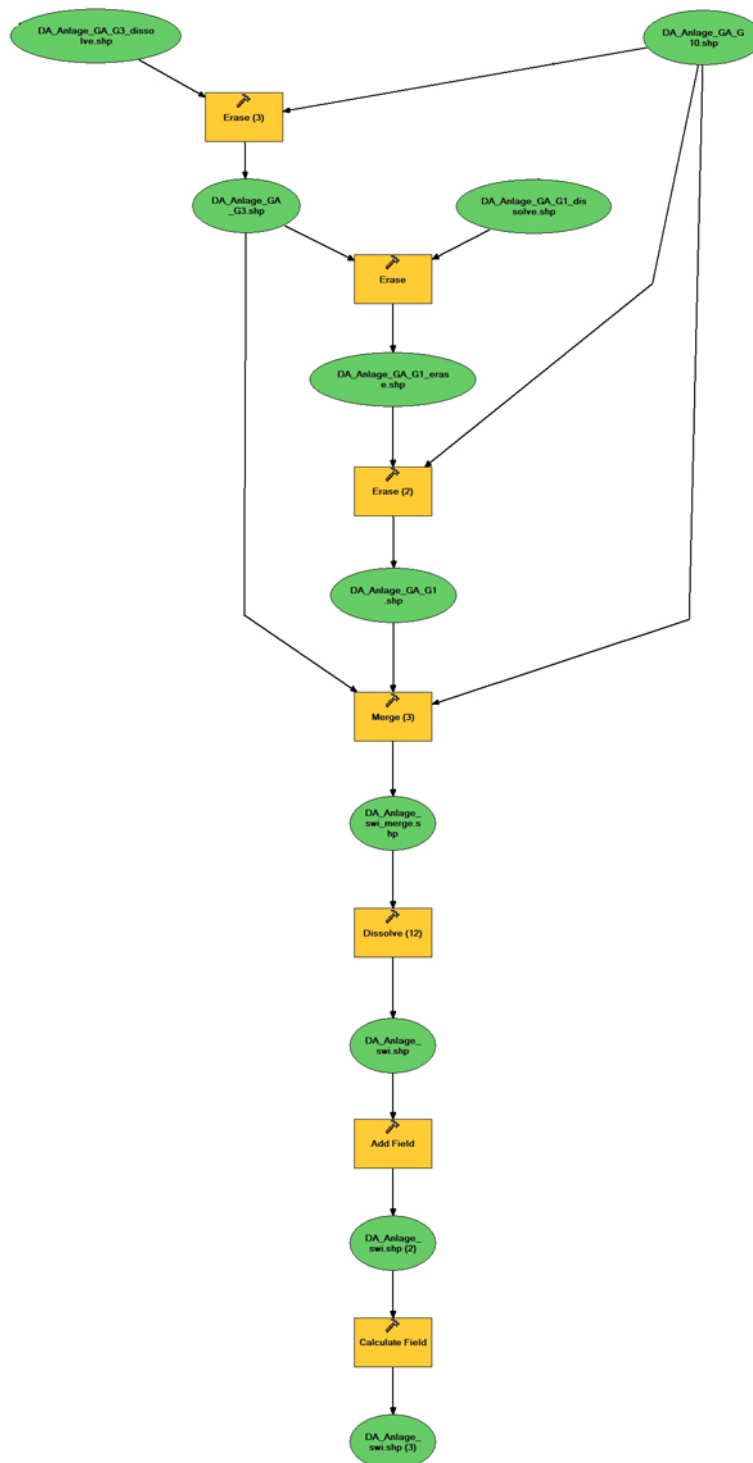


Modèle 8: Modèle pour les installations (partie 3).

#### Détails du modèle

##### GA G10:

- **V25\_anl\_polygon:** **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked



Modèle 9: Modèle pour les installations (partie 4).

#### Détails du modèle

- **Erase (3):** Input Features: DA\_Anlage\_GA\_G3\_dissolve, Erase Features: DA\_Anlage\_GA\_G10
- **Erase:** Input Features: DA\_Anlage\_GA\_G1\_dissolve, Erase DA\_Anlage\_Ga\_G3
- **Erase (2):** Input Features: DA\_anlage\_GA\_G1\_erase, Erase Features: DA\_Anlage\_GA\_G10
- **Merge:** DA\_Anlage\_GA\_G10, DA\_Anlage\_GA\_G3 et DA\_Anlage\_GA\_G1
- **Dissolve:** Create Multipart Feature: non checked
- **Add Field:** Field Name: DA\_AnI, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_AnI, Expression: 1



### 4.3 Les bâtiments

Les bâtiments sont divisés en quatre catégories distinctes:

- Maisons d'habitation habitées en permanence (chapitre 4.3.1)
- Maisons d'habitation habitées périodiquement (chapitre 4.3.2)
- Bâtiments industriels (chapitre 4.3.3) divisés en sous catégories:
  - o Industries sous forme de point (chapitre 4.3.3.1), industries définies à partir du recensement des entreprises (EZ), ne se trouvant pas sur la donnée "bâtiment" du VECTOR25 et n'appartenant pas aux catégories "installations" ou "bâtiments publics".
  - o Petites industries (chapitre 4.3.3.2), industries définies à partir du recensement des entreprises (EZ), se trouvant sur la donnée "bâtiment" du VECTOR25 et n'appartenant pas aux catégories "installations" ou "bâtiments publics". Seuls les bâtiments ayant une surface inférieure à 1963 m<sup>2</sup> (rayon 25 m) sont considérés. Dans ce cas-là, la surface considérée sera le point du recensement avec une zone tampon de 25 m et non pas la surface du bâtiment.
  - o Grandes industries (chapitre 4.3.3.3), industries définies à partir du recensement des entreprises (EZ), se trouvant sur la donnée "bâtiment" du VECTOR25 et n'appartenant pas aux catégories "installations" ou "bâtiments publics". Seuls les bâtiments ayant une surface supérieure à 1963 m<sup>2</sup> (rayon 25 m) sont considérés.
  - o Surfaces industrielles (chapitre 4.3.3.4), bâtiments de la donnée "bâtiment" du VECTOR 25, pour autant qu'ils ne fassent pas partie des catégories "installations" ou "bâtiments publics" et qui appartiennent aux zones industrielles de la donnée "utilisation du sol 1997".
- Bâtiments publics (chapitre 4.3.4)

#### Description générale des modèles:

Les modèles décrits dans les chapitre 4.3.1 à 4.3.4 sont subdivisés selon trois types de pondération:

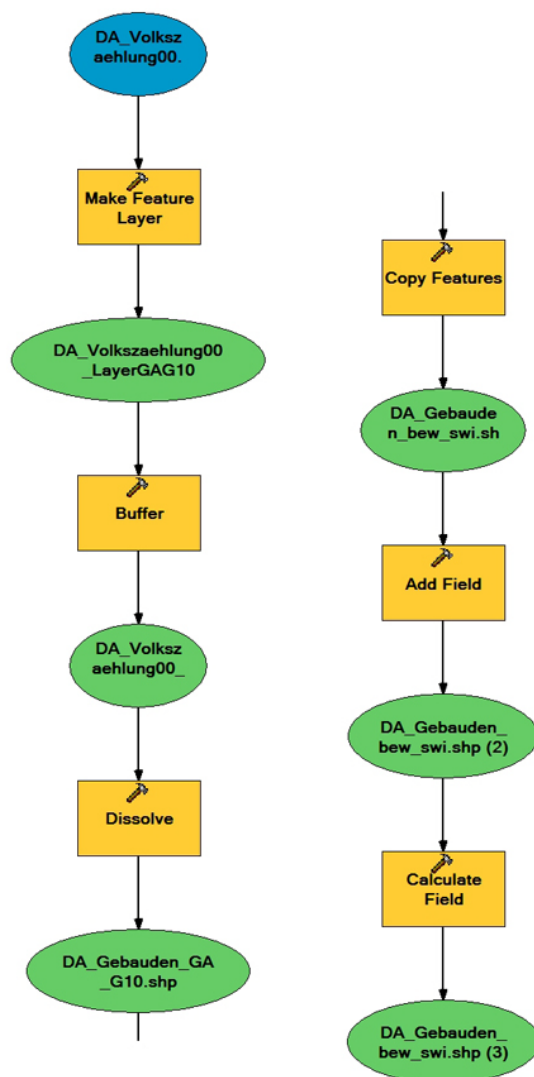
- **Pondération 3:** Station d'arrêt avec un faible degré de probabilité
  - o Objets: bâtiments publiques comme église, perron (halle de quais), château, station et arrêt de transports publics
- **Pondération 5:** Station d'arrêt avec un haut degré de probabilité
  - o Objets: maison d'habitation habitée périodiquement, bâtiment industriel et artisanal
- **Pondération 10:** Présence humaine durable ou partiellement durable
  - o Objets: maison d'habitation habitée en permanence

#### 4.3.1 Maisons d'habitation habitées en permanence

**Modèle:** Gebäude GA10 (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Volkszaehlung00

**Output:** DA\_Gebauden\_GA\_G10 et DA\_Gebauden\_bew\_swi.



Modèle 10: Modèle pour les habitations permanentes.

##### Détails du modèle

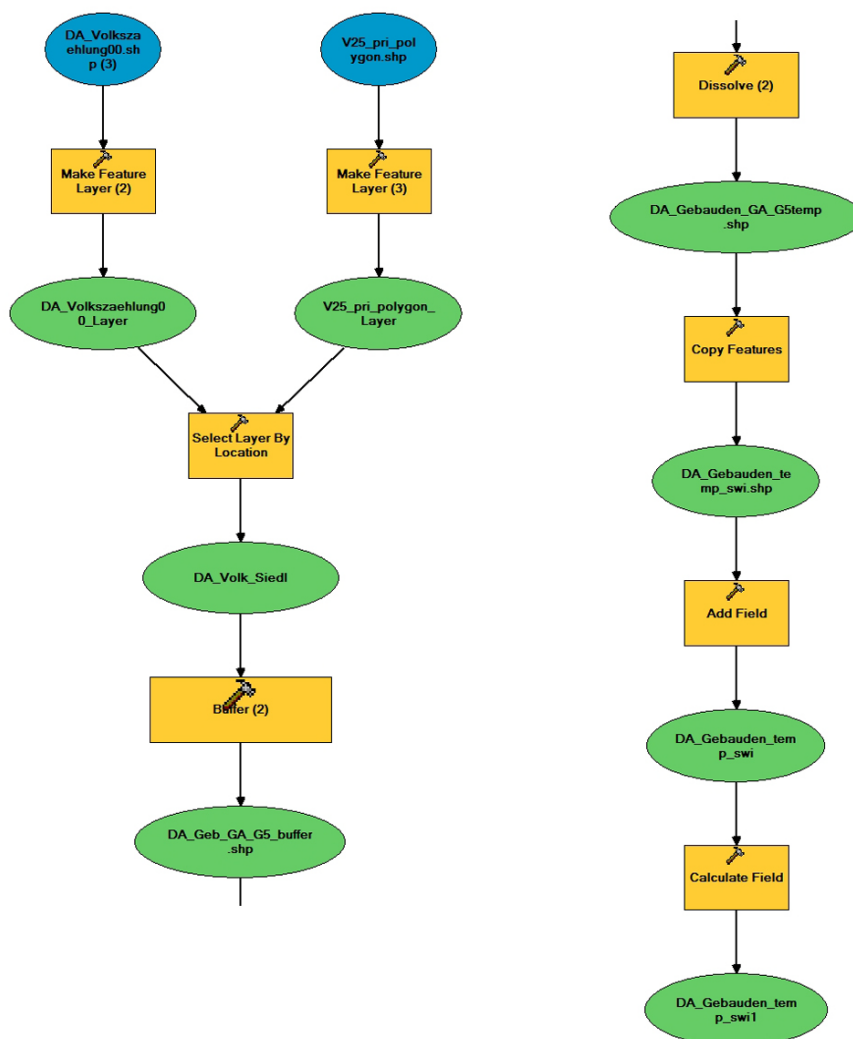
- **DA\_Volkszaehlung00:** Make Feature Layer: A00WDTOT > 0
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Copy Feature:** DA\_Gebauden\_GA\_G10
- **Add Field:** Field Name: DA\_Geb\_bew, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_Geb\_bew, Expression: 1

### 4.3.2 Maisons d'habitation habitées périodiquement

**Modèle:** Gebäude GA5 temp (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Volkszaehlung00 et V25\_pri\_polygon

**Output:** DA\_Gebauden\_GA\_G5temp et DA\_Gebauden\_temp\_swi.



Modèle 11: Modèle pour les habitations périodiquement.

#### Détails du modèle

- **V25\_pri\_polygon:** Make Feature Layer: "OBJEKTVAL = 'Z\_Siedl'"
- **DA\_Volkszaehlung00:** Make Feature Layer: "A00WDTOT " 0 AND "A00WTZTOT" > 0
- **Select Layer By Location:** Input: DA\_Volkszaehlung\_Layer, Selecting: V25\_pri\_polygon\_Layer
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Copy Feature:** DA\_Gebauden\_GA\_G5temp
- **Add Field:** Field Name: DA\_Geb\_bew, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_Geb\_bew, Expression: 1

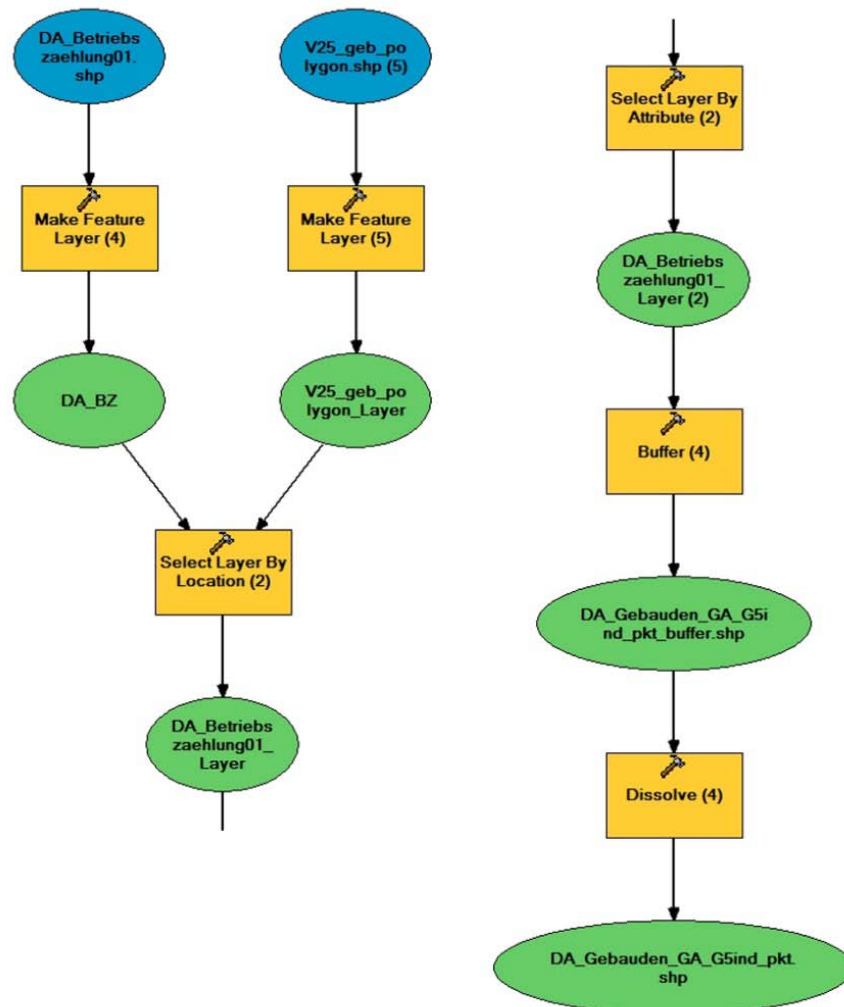
### 4.3.3 Bâtiments industriels

#### 4.3.3.1 Industries sous forme de point

**Modèle:** Gebäude GA5 ind punkt (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Betriebszaehlung et V25\_geb\_polygon

**Output:** DA\_Gebauden\_GA\_G5ind\_pkt.



Modèle 12: Modèle pour les industries sous forme de point.

#### Détails du modèle

- **DA\_Betriebszaehlung01:** Make Feature Layer
- **V25\_geb\_polygon:** Make Feature Layer: "OBJEKTVAL = 'Z\_Gasthof' OR "OBJEKTVAL = 'Z\_Gebaeude' OR "OBJEKTVAL = 'Z\_Huette' OR "OBJEKTVAL = 'Z\_Schiessstand' OR "OBJEKTVAL = 'Z\_Treibhaus' OR "OBJEKTVAL = 'Z\_Innenhof"
- **Select Layer By Location:** Input: DA\_BZ, Selecting: V25\_geb\_polygon\_Layer
- **Select Layer By Attribute:** Layer Name: DA\_Betriebszaehlung01\_Layer, Selection type: SWITCH SELECTION
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked

#### 4.3.3.2 Petites industries

**Modèle:** Gebäude GA5 ind klein (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Betriebezaehlung et V25\_geb\_polygon

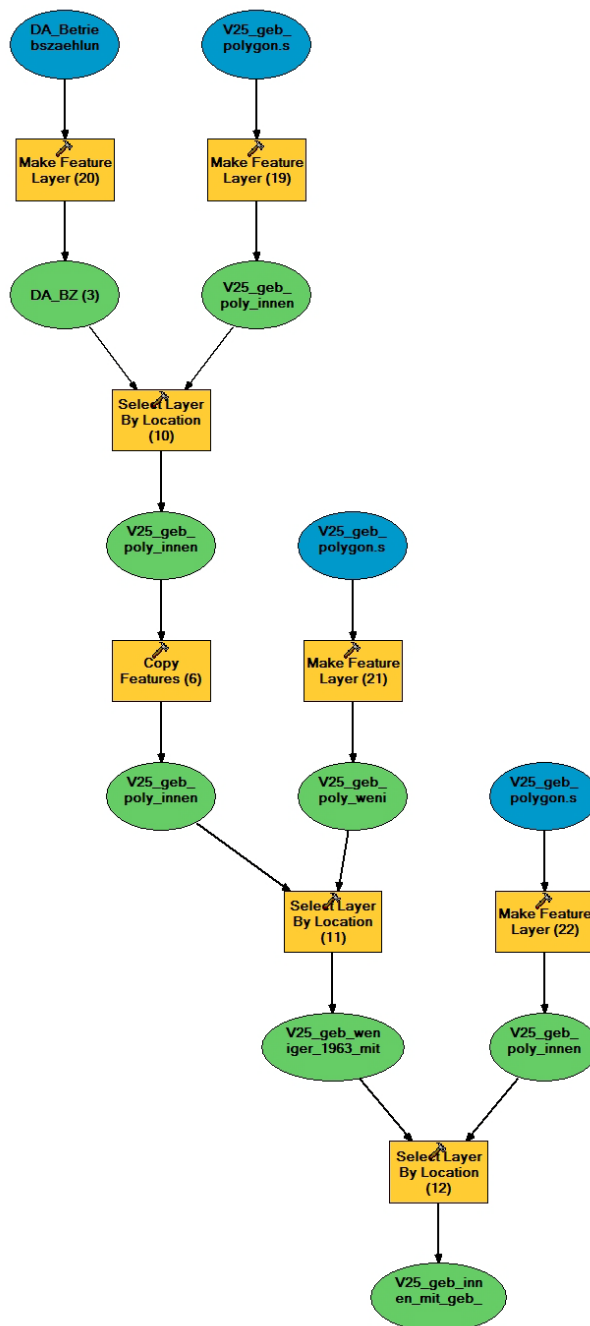
**Output:** DA\_Gebauden\_GA\_G5ind\_klein.

##### Remarque sur le problème dû aux cours intérieures:

Il arrive parfois que le point (coordonnées X et Y) du recensement des entreprises se trouve dans la cour intérieure d'un bâtiment et non sur le bâtiment lui-même. Afin de résoudre ce problème, on va d'abord sélectionner la cour intérieure du bâtiment et après analyser la taille du bâtiment en déduisant la surface de sa cour intérieure. Cette étape est directement intégrée dans le modèle.



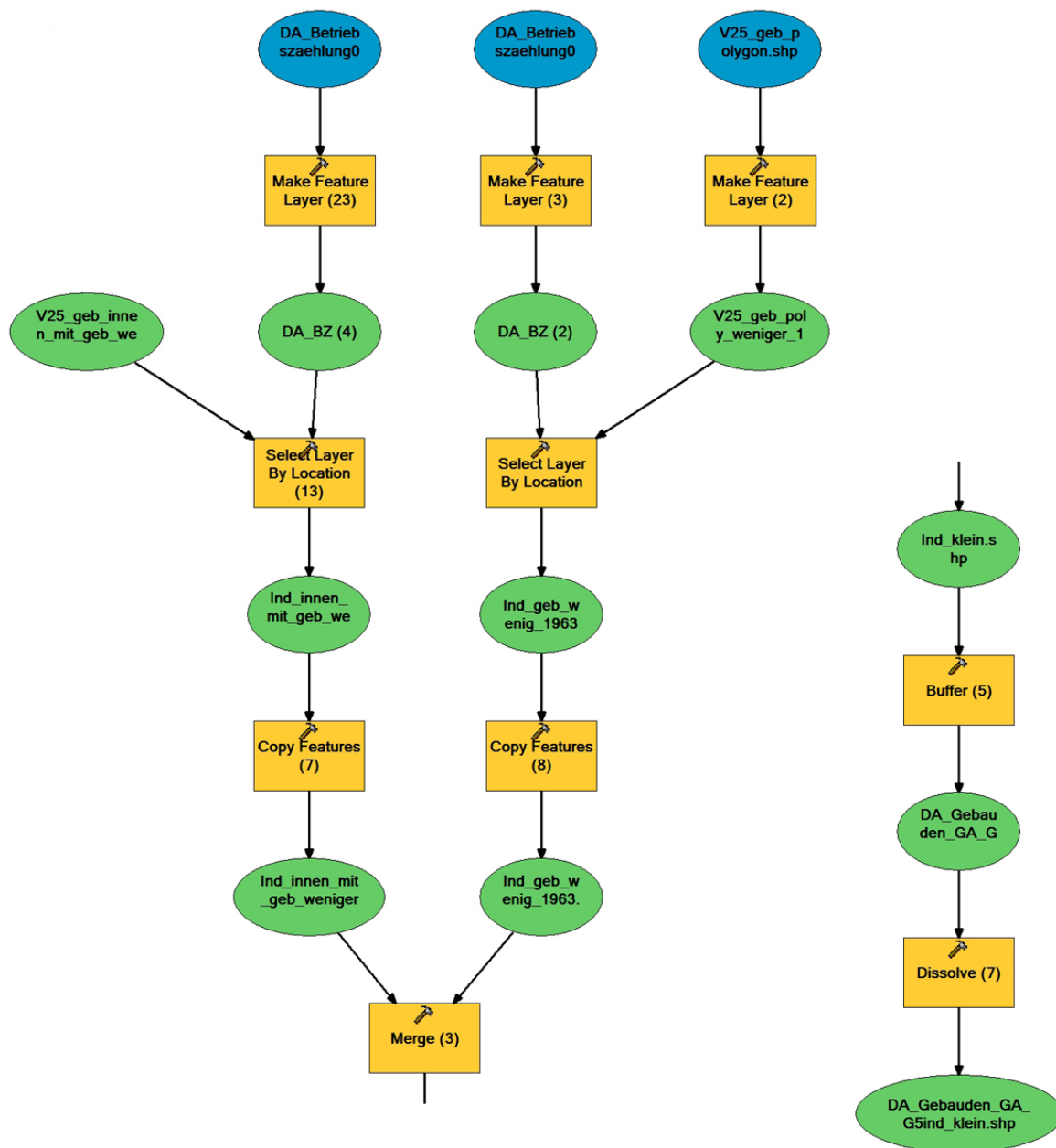
Figure 24: Points du recensement dans la cour intérieur.



Modèle 13: Petites industries dérivant des cours intérieures.

### Détails du modèle

- **DA\_Betriebszaehlun01: Make Feature Layer**
- **V25\_geb\_polygon: Make Feature Layer:** "OBJEKTVAL" = 'Z\_Innenhof'
- **Select Layer By Location:** Input: V25\_geb\_poly\_innen, Selecting: DA\_BZ (3)
- **Copy Feature:** Input: V25\_geb\_poly\_innen\_mit\_BZ
- **V25\_geb\_polygon: Make Feature Layer:** ("OBJEKTVAL" = 'Z\_Gasthof' OR "OBJEKTVAL" = 'Z\_Gebaeude' OR "OBJEKTVAL" = 'Z\_Huette' OR "OBJEKTVAL" = 'Z\_Schiessstand' OR "OBJEKTVAL" = 'Z\_Treibhaus') AND "AREA" <= 1963,5
- **Select Layer By Location:** Input: V25\_geb\_poly\_weniger\_1963 (2), Selecting: V25\_geb\_poly\_innen\_mit\_BZ
- **V25\_geb\_polygon: Make Feature Layer:** "OBJEKTVAL" = 'Z\_Innenhof'
- **Select Layer By Location:** Input: V25\_geb\_poly\_innen (2), Selecting: V25\_geb\_weniger\_1963\_mit\_innen



Modèle 14: Suite du modèle pour les petites industries.

### Détails du modèle

- **DA\_Betriebezaehlung01: Make Feature Layer**
- **Select Layer By Location:** Input: DA\_BZ (4), Selecting: V25\_geb\_innen\_mit\_geb\_weniger\_1963
- **Copy Feature:** Input: Ind\_innen\_mit\_geb\_weniger\_1963
- **DA\_Betriebezaehlung01: Make Feature Layer**
- **V25\_geb\_polygon: Make Feature Layer:** ("OBJEKTVAL" = 'Z\_Gasthof' OR "OBJEKTVAL" = 'Z\_Gebaeude' OR "OBJEKTVAL" = 'Z\_Huette' OR "OBJEKTVAL" = 'Z\_Schiessstand' OR "OBJEKTVAL" = 'Z\_Treibhaus') AND "AREA" <= 1963,5
- **Select Layer By Location:** Input: DA\_BZ (2), Selecting: V25\_geb\_poly\_weniger\_1963
- **Copy Feature:** Input: Ind\_geb\_wenig\_1963
- **Merge:** Input: Ind\_innen\_mit\_geb\_weniger\_1963.shp et Ind\_geb\_wenig\_1963
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked

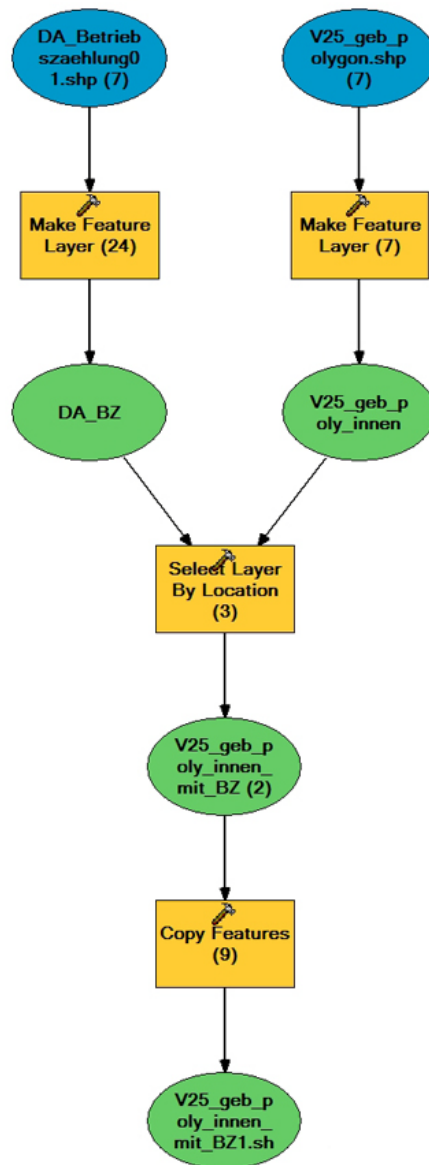


#### 4.3.3.3 Grandes Industries

**Modèle:** Gebäude GA5 ind gross (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Betriebezaehlung et V25\_geb\_polygon

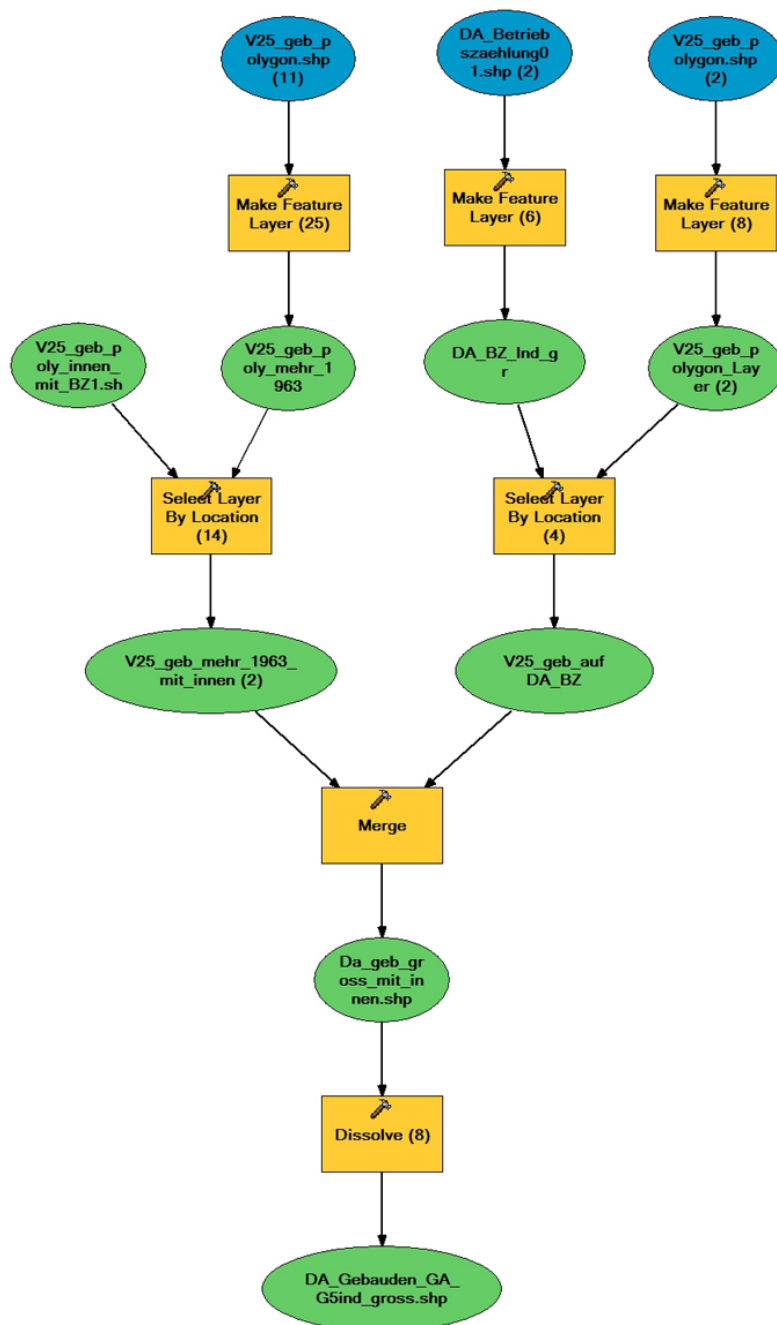
**Output:** DA\_Gebauden\_GA\_G5ind\_gross.



Modèle 15: Cours intérieures pour les grandes industries.

##### Détails du modèle

- **DA\_Betriebezaehlung01:** Make Feature Layer
- **V25\_geb\_polygon:** Make Feature Layer: "OBJEKTVAL" = 'Z\_Innenhof'
- **Select Layer By Location:** Input: V25\_geb\_poly\_innen, Selecting: DA\_BZ
- **Copy Feature:** Input: V25\_geb\_poly\_innen\_mit\_BZ



Modèle 16: Suite pour les grandes industries.

#### Détails du modèle

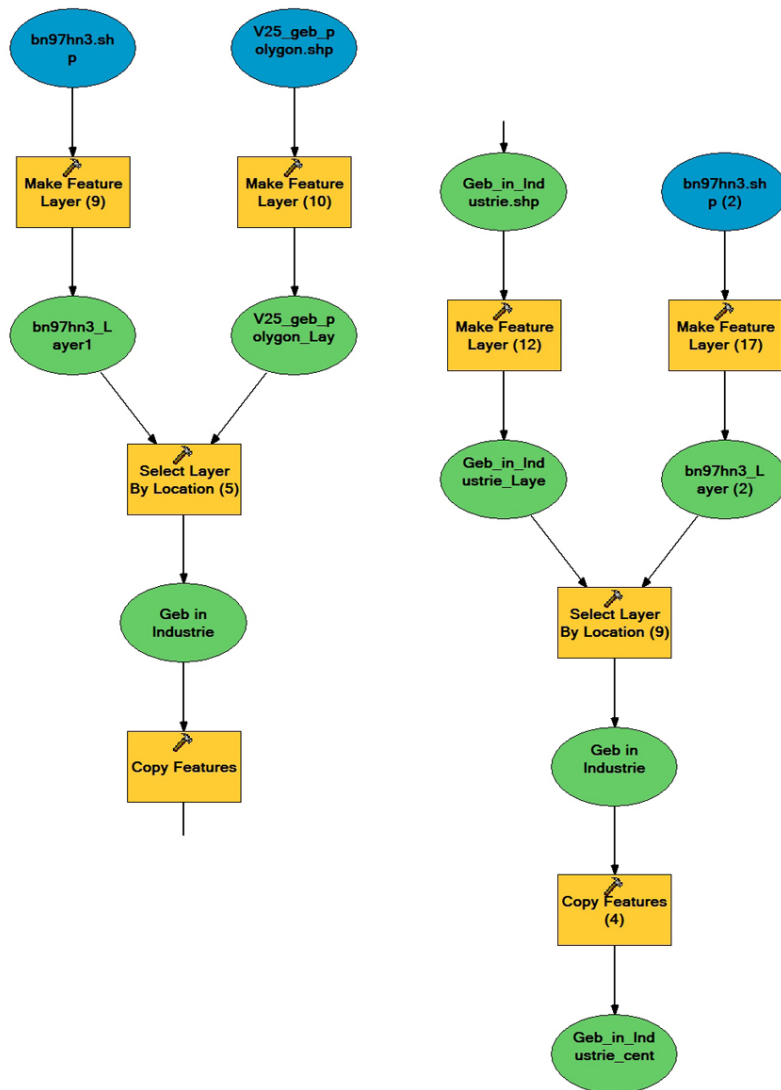
- **V25\_geb\_polygon: Make Feature Layer:** ("OBJEKTVAL" = 'Z\_Gasthof' OR "OBJEKTVAL" = 'Z\_Gebaeude' OR "OBJEKTVAL" = 'Z\_Huette' OR "OBJEKTVAL" = 'Z\_Schiessstand' OR "OBJEKTVAL" = 'Z\_Treibhaus') AND "AREA" > 1963,5
- **Select Layer By Location:** Input: V25\_geb\_poly\_mehr\_1963, Selecting: V25\_geb\_poly\_innen\_mit\_BZ1
- **DA\_Betriebszaehlung01: Make Feature Layer**
- **V25\_geb\_polygon: Make Feature Layer:** ("OBJEKTVAL" = 'Z\_Gasthof' OR "OBJEKTVAL" = 'Z\_Gebaeude' OR "OBJEKTVAL" = 'Z\_Huette' OR "OBJEKTVAL" = 'Z\_Schiessstand' OR "OBJEKTVAL" = 'Z\_Treibhaus') AND "AREA" > 1963,5
- **Select Layer By Location:** Input: V25\_geb\_polygon\_Layer (2), Selecting: DA\_BZ\_Ind\_gr
- **Merge:** Input: V25\_geb\_auf DA\_BZ et V25\_geb\_mehr\_1963\_mit\_innen (2)
- **Dissolve:** Create Multipart Feature: non checked

#### 4.3.3.4 Surfaces industrielles

**Modèle:** Gebäude GA5 ind areal (Toolbox: SiPro 93 DAMAGE)

**Input:** bn97hn3, DA\_Betriebezaehlung, DA\_Volkzaehlung00 et V25\_geb\_polygon

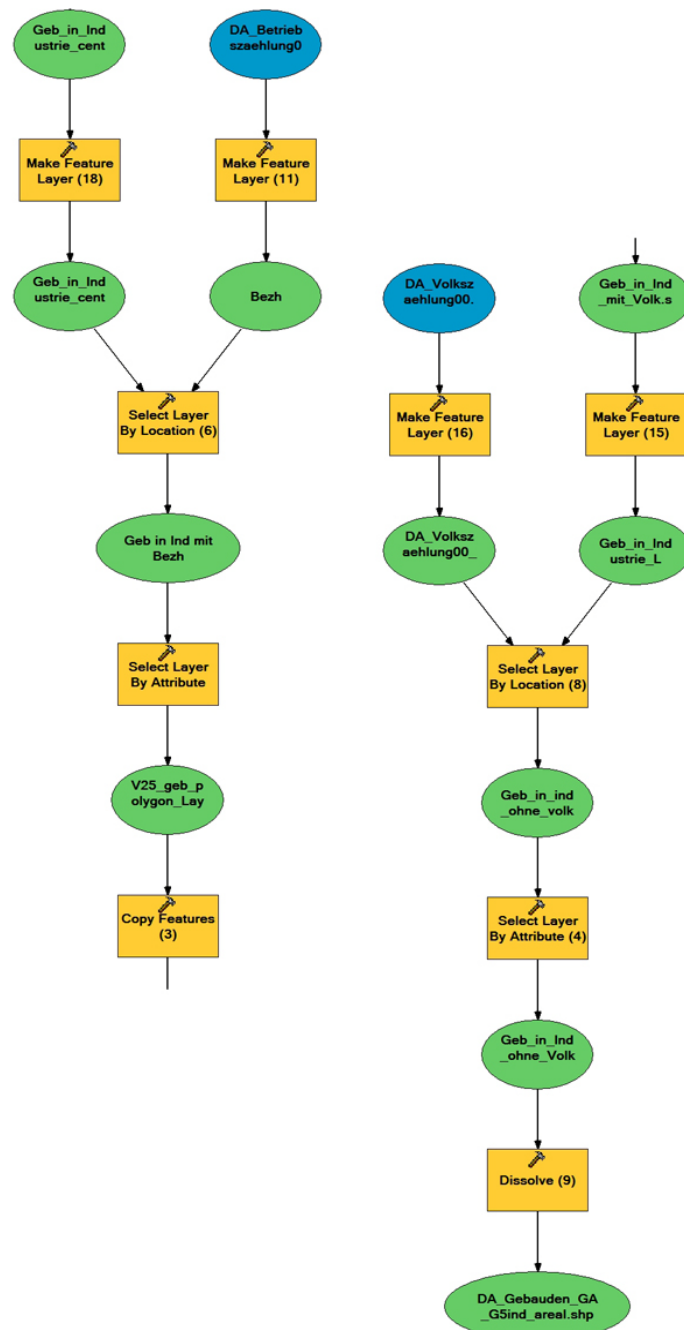
**Output:** DA\_Gebauden\_GA\_G5ind\_areal.



Modèle 17: Surfaces industrielles (1<sup>ère</sup> partie).

#### Détails du modèle

- **Bn97hn3:** Make Feature Layer: "GRIDCODE" = 15
- **V25\_geb\_polygon:** Make Feature Layer: "OBJEKTVAL" = 'Z\_Gasthof' OR "OBJEKTVAL" = 'Z\_Gebaeude' OR "OBJEKTVAL" = 'Z\_Huette' OR "OBJEKTVAL" = 'Z\_Schiessstand' OR "OBJEKTVAL" = 'Z\_Treibhaus'
- **Select Layer By Location:** Input: V25\_geb\_polygon\_Layer, Selecting: bn97hn3\_Layer1
- **Copy Feature:** Input: Geb\_in\_Industrie
- **Geb\_in\_Industrie:** Make Feature Layer
- **Bn97hn3:** Make Feature Layer: "GRIDCODE" = 15
- **Select Layer By Location:** Input: Geb\_in\_Industrie\_Layer, Relationship: HAVE\_THEIR\_CENTRE\_IN Selecting: bn97hn3\_Layer (2)
- **Copy Feature:** Input: Geb in Industrie (2)



Modèle 18: Surfaces industrielles (2<sup>ème</sup> partie).

#### Détails du modèle

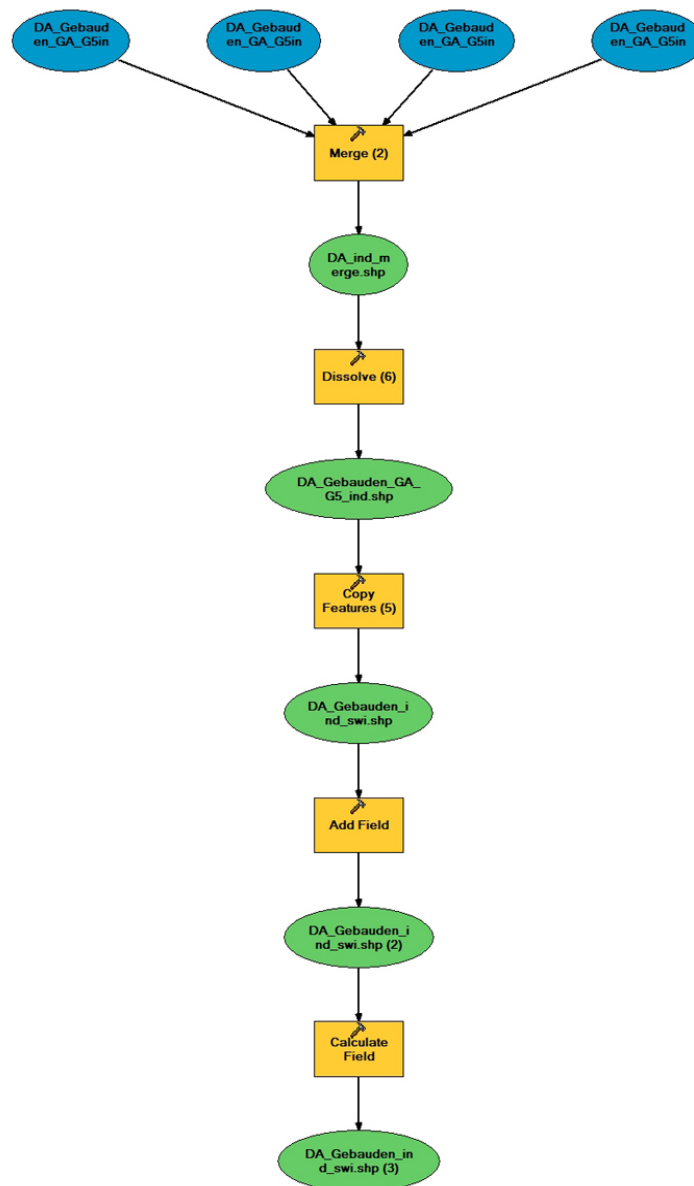
- **Geb\_in\_Industrie\_center**: Make Feature Layer
- **DA\_Betriebszaehlung01**: Make Feature Layer
- **Select Layer By Location**: Input: Geb\_in\_Industrie\_center\_Layer, Selecting: Bezh
- **Select Layer By Attribute**: Layer Name: Geb in Ind mit Bezh, Selection type: SWITCH\_SELECTION
- **Copy Feature**: V25\_geb\_polygon\_Layer2 (2)
- **DA\_Volkszaehlung00\_Layer**: Make Feature Layer
- **Geb\_in\_Ind\_mit\_Volk**: Make Feature Layer
- **Select Layer By Location**: Input: Geb\_in\_Industrie\_L, Selecting: DA\_VolksZaehlung00\_Layer1
- **Select Layer By Attribute**: Layer Name: Geb\_in\_Ind\_ohne\_Volk, Selection type: SWITCH\_SELECTION
- **Dissolve**: Create Multipart Feature: non checked

#### 4.3.3.5 Combinaison des différents types d'industries

**Modèle:** Gebäude GA5 ind (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_gebauden\_GA5 ind pkt, DA\_gebauden\_GA5 ind klein, DA\_gebauden\_GA5 ind\_gross et DA\_gebauden\_GA5 ind areal

**Output:** DA\_Gebauden\_GA\_G5ind et DA\_Gebauden\_ind\_swi.



Modèle 19: Regroupement des industries de pondération 5.

##### Détails du modèle

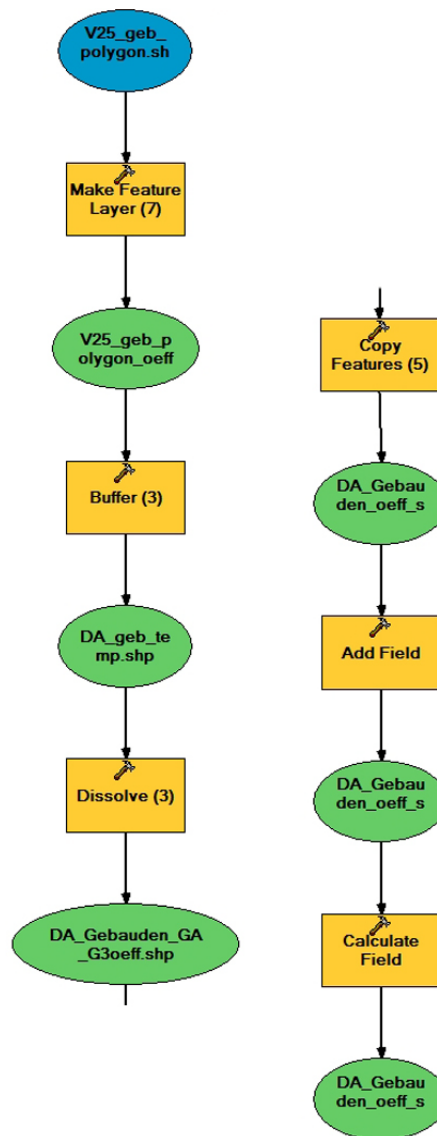
- **Merge:** Input: DA\_gebauden\_GA5 ind pkt, DA\_gebauden\_GA5 ind klein, DA\_gebauden\_GA5 ind gross und DA\_gebauden\_GA5 ind areal
- **Dissolve:** Create Multipart Feature: non checked
- **Copy Feature:** Input: DA\_Gebauden\_GA\_G5\_ind
- **Add Field:** Field Name: DA\_ind, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_ind, Expression: 1

#### 4.3.4 Bâtiments publics

Modèle: Gebäude GA3 oeff (Toolbox: SiPro 93 DAMAGE)

Input: V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G3oeff et DA\_Gebauden\_oeff\_swi.



Modèle 20: Bâtiments publics.

##### Détails du modèle

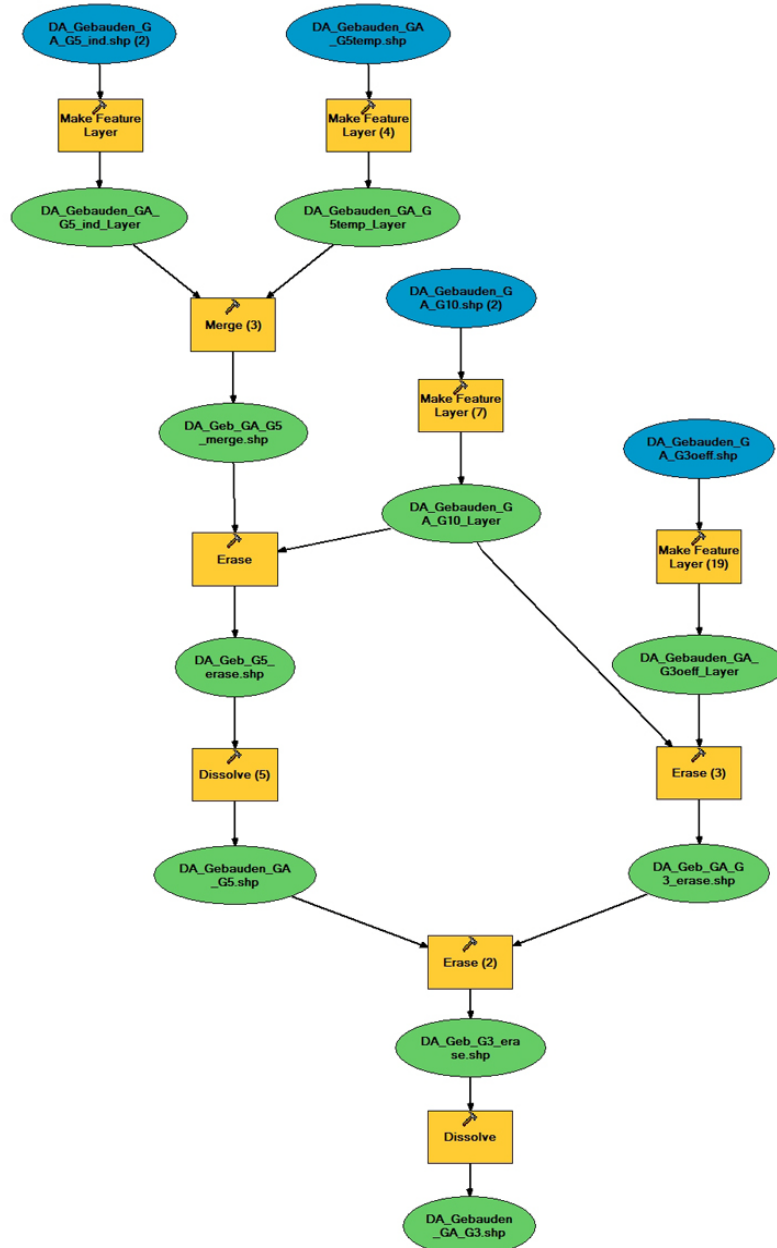
- **V25\_geb\_polygon:** Make Feature Layer: "OBJEKTVAL = 'Z\_Kirche' OR "OBJEKTVAL = 'Z\_Perron' OR "OBJEKTVAL = 'Z\_Schloss' OR "OBJEKTVAL = 'Z\_Station'"
- **Buffer:** Distance: 25m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Copy Feature:** DA\_Gebauden\_GA\_G5oeff
- **Add Field:** Field Name: DA\_Geb\_bew, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_Geb\_bew, Expression: 1

#### 4.3.5 Regroupement des bâtiments de pondérations 3 et 5

**Modèle:** Gebäude GA5 GA3 (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Gebauden\_GA\_G5ind, DA\_Gebauden\_GA\_G5temp, DA\_Gebauden\_GA\_G3\_oeff et DA\_Gebauden\_GA\_G10

**Output:** DA\_Gebauden\_GA\_G5 et DA\_Gebauden\_GA\_G3.



Modèle 21: Bâtiments de pondérations 3 et 5.

##### Détails du modèle

- **Erase:** Input: DA\_GEB\_GA\_G5\_merge, Erase: DA\_Gebauden\_GA\_G10\_Layer
- **Dissolve:** Create Multipart Feature: non checked
- **Erase (3):** Input: DA\_Gebauden\_GA\_G5oeff\_Layer, Erase: DA\_Gebauden\_GA\_G10\_Layer
- **Dissolve:** Create Multipart Feature: non checked
- **Erase (2):** Input: DA\_Geb\_GA\_G3\_erase, Erase: DA\_Gebauden\_GA\_G5\_Layer
- **Dissolve:** Create Multipart Feature: non checked



#### 4.4 Le réseau routier

**Modèle:** Strassen GA (Toolbox: SiPro 93 DAMAGE)

**Input:** Str\_25\_arc.

**Output:** DA\_Strasse\_GA\_G5, DA\_Strasse\_GA\_G3, DA\_Strassen\_swi et DA\_Autobahn\_swi.

##### Description du modèle:

Ce modèle est subdivisé selon deux types de pondération:

- **Pondération 3:** Station d'arrêt avec un faible degré de probabilité
  - o Objets: routes de 1<sup>ère</sup> et 2<sup>ème</sup> classes
  - o Remarque: pas de tunnel
  
- **Pondération 5:** Station d'arrêt avec un haut degré de probabilité
  - o Objets: autoroute, chaussée séparée des autoroutes, semi-autoroute, accès d'auto-route, entrée et sortie d'autoroute
  - o Remarque: pas de tunnel



Modèle 22: Routes.

### Détails du modèle

- **Str\_25\_arc: Make Feature Layer:** ("OBJECTVAL" = 'Autobahn' OR "OBJECTVAL" = 'Autob\_Ri' OR "OBJECTVAL" = 'Autostr' OR "OBJECTVAL" = 'Ein\_Ausf' OR "OBJECTVAL" = 'A\_Zufahrt') AND "TUNNELTYPE" = " "
- **Buffer:** Distance: 10m, End Type: ROUND
- **Dissolve:** Create Multipart Feature: non checked
- **Add Field:** Field Name: DA\_Str, Field Type: SHORT
- **Calculate Field:** Field Name: DA\_Str, Expression: 1
- **Str\_25\_arc: Make Feature Layer (2):** ("OBJECTVAL" = '1\_Klass AND "TUNNELTYPE" = " "
- **Buffer (2):** Distance: 10m, End Type: ROUND
- **Dissolve (2):** Create Multipart Feature: non checked
- **Add Field (2):** Field Name: DA\_Autob, Field Type: SHORT
- **Calculate Field (2):** Field Name: DA\_Autob, Expression: 1
- **Erase:** Input Features: DA\_Strassen\_swi, Erase Features: DA\_Autobahn\_swi
- **Copy Feature:** Input Features: DA\_Autobahn\_swi

## 4.5 Combinaison des enjeux pour le calcul des indices

### 4.5.1 Enjeux pour l'indice de forêts protectrices

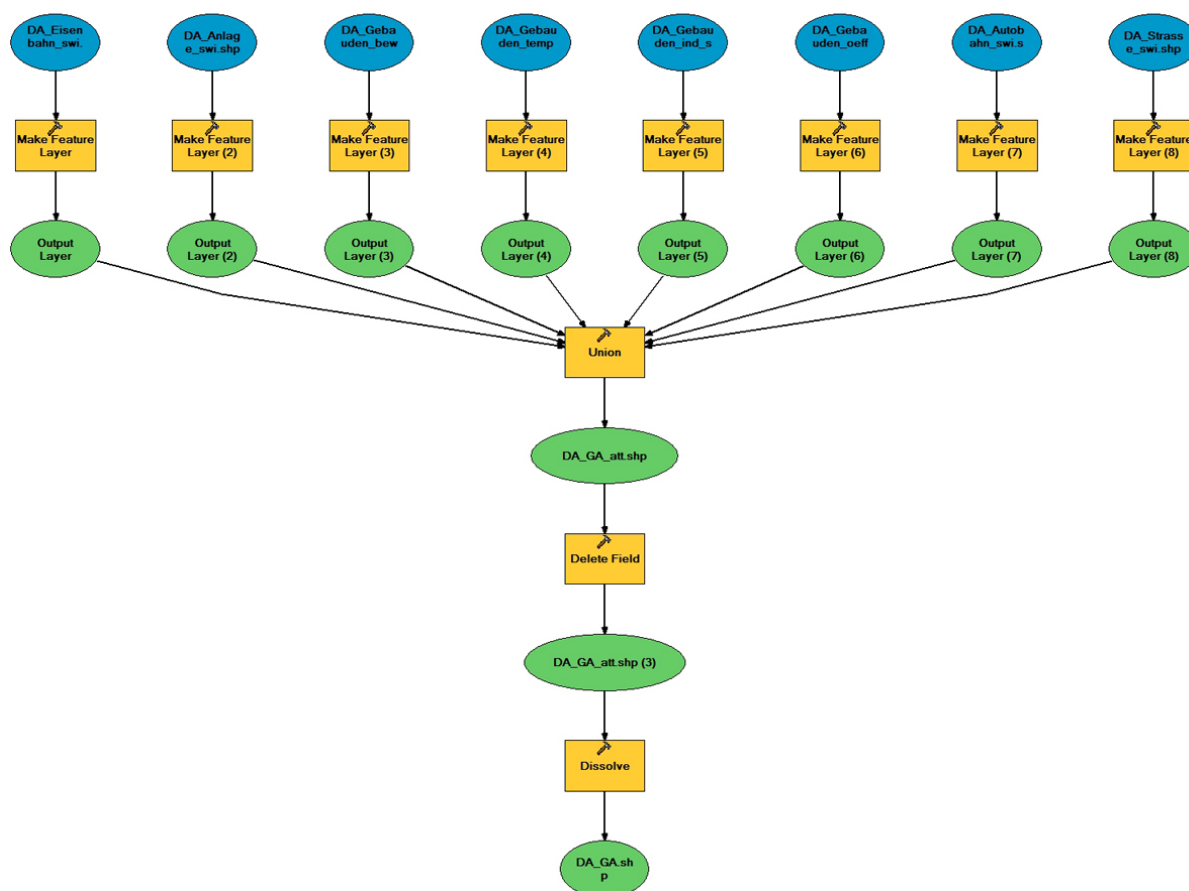
**Modèle:** DA\_GA (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Eisenbahn\_swi, DA\_Anlage\_swi, DA\_Gebauden\_bew\_swi, DA\_Gebauden\_temp\_swi, DA\_Gebauden\_ind\_swi, DA\_Gebauden\_oeff\_swi, DA\_Autobahn\_swi et DA\_Strassen\_swi.

**Output:** DA\_GA\_att et DA\_GA.

#### Description du modèle:

Toutes les données calculées dans les chapitres précédents sont maintenant mises ensembles.



Modèle 23: Enjeux pour le calcul de l'indice de forêts protectrices.

#### Détails du modèle

- **Make Feature Layer** DA\_Eisenbahn\_swi, DA\_Anlage\_swi, DA\_Gebauden\_bew\_swi, DA\_Gebauden\_temp\_swi, DA\_Gebauden\_ind\_swi, DA\_Gebauden\_oeff\_swi, DA\_Autobahn\_swi et DA\_Strassen\_swi
- **Union:** DA\_Eisenbahn\_swi, DA\_Anlage\_swi, DA\_Gebauden\_bew\_swi, DA\_Gebauden\_temp\_swi, DA\_Gebauden\_ind\_swi, DA\_Gebauden\_oeff\_swi, DA\_Autobahn\_swi und DA\_Strassen\_swi
- **Delete Field:** Drop Field: FID\_DA\_Eis, Id, FID\_DA\_AnI, Id\_1, FID\_DA\_Geb\_Id\_12, FID\_DA\_G\_1, Id\_12\_13, FID\_DA\_G\_2, Id\_12\_1\_14, FID\_DA\_Aut, Id\_12\_1\_15, FID\_DA\_Str, Id\_12\_1\_16 et Id\_12\_1\_17
- **Dissolve:** Create Multipart Feature: non checked

Dans le tableau des attributs de la donnée **DA\_GA\_att**, tous les types d'enjeux (réseau ferroviaire, installation, bâtiment et réseau routier) ont une valeur de 1 s'ils sont présents et une valeur de 0 s'il y en a pas; ainsi en effectuant la bonne sélection, il est possible de choisir le type d'enjeux voulu.

| FID | Shape   | DA_Eis | DA_Geb_bew | DA_Geb_tmp | DA_Ind | DA_Geb_oef | DA_Autob | DA_Str | DA_Anl |
|-----|---------|--------|------------|------------|--------|------------|----------|--------|--------|
| 0   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 1   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 2   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 3   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 4   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 5   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 6   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 7   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 8   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 9   | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 10  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 11  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 12  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 13  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 14  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 15  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 16  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 17  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 18  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 19  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 20  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 21  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 22  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 23  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 24  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 25  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 26  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 27  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |
| 28  | Polygon | 1      | 0          | 0          | 0      | 0          | 0        | 0      | 0      |

Tableau 5: Attributs de la donnée géo-référencée "enjeu".

La donnée géo-référencée "enjeu" a été divisée en 49 carrés de 40 kilomètres (Figure 16) afin d'augmenter la vitesse de calcul des différents modèles de géotraitement.

#### 4.5.2 Enjeux utilisés pour le calcul de l'indice du potentiel de dommage

**Modèle:** DA\_Gew\_Kt\_T1 (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Eisenbahn\_GA\_G1, DA\_Eisenbahn\_GA\_G3, DA\_Eisenbahn\_GA\_G5, DA\_Anlage\_GA\_G1, DA\_Anlage\_GA\_G3, DA\_Anlage\_GA\_G10, DA\_Gebauden\_GA\_G3, DA\_Gebauden\_GA\_G5, DA\_Gebauden\_GA\_G10, DA\_Strassen\_GA\_G3 et DA\_Strassen\_GA\_G5.

**Output:** DA\_GA\_10\_Diss, DA\_GA\_5\_Diss, DA\_GA\_3\_Diss et DA\_GA\_1\_Diss.

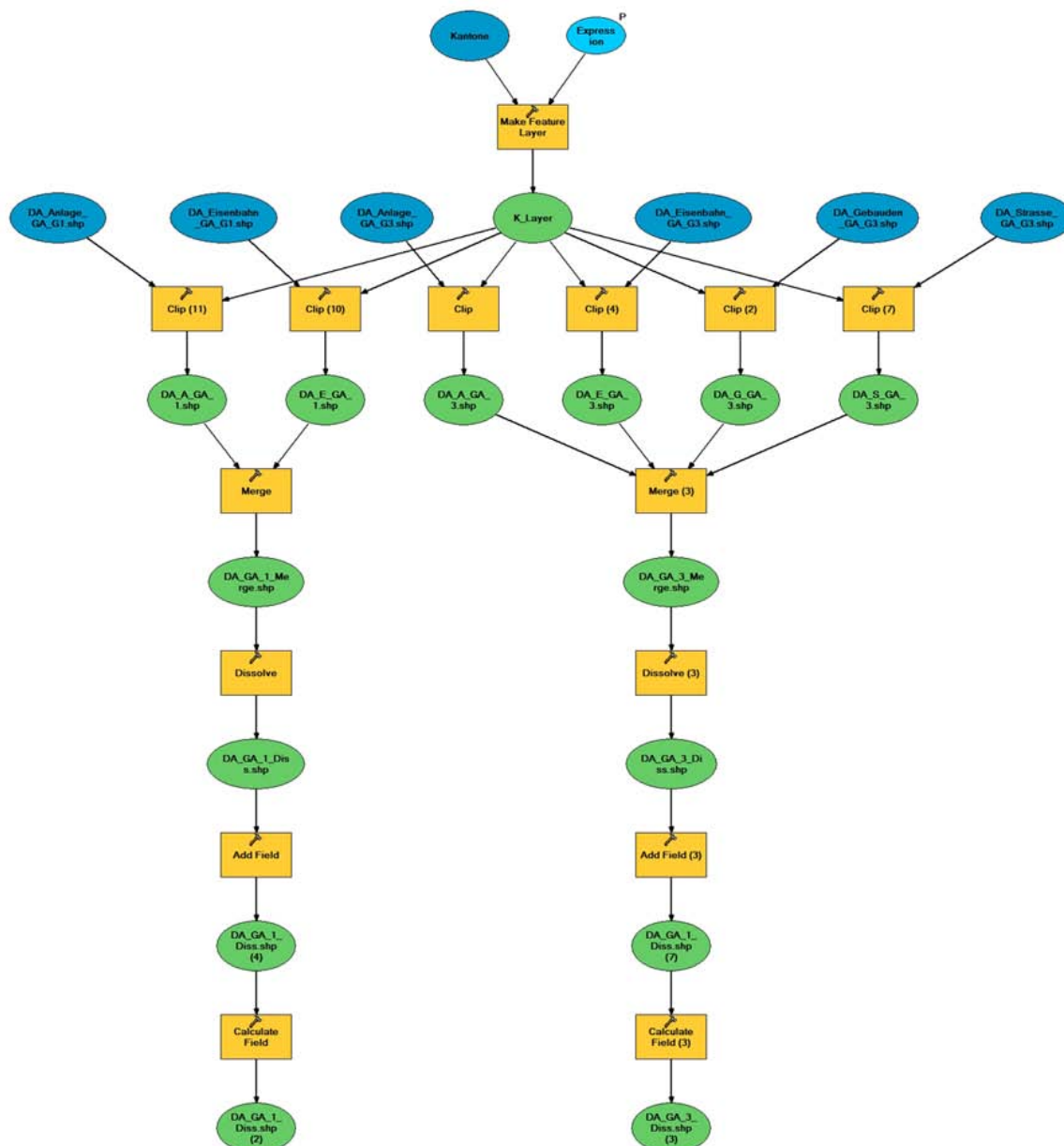
##### Description du modèle

Ce modèle différencie les enjeux selon leur pondération (Tableau 3) et selon le canton.



Figure 25: Pondération de l'enjeu: 10 (rose), 5 (vert), 3 (jaune) et 1 (bleu).

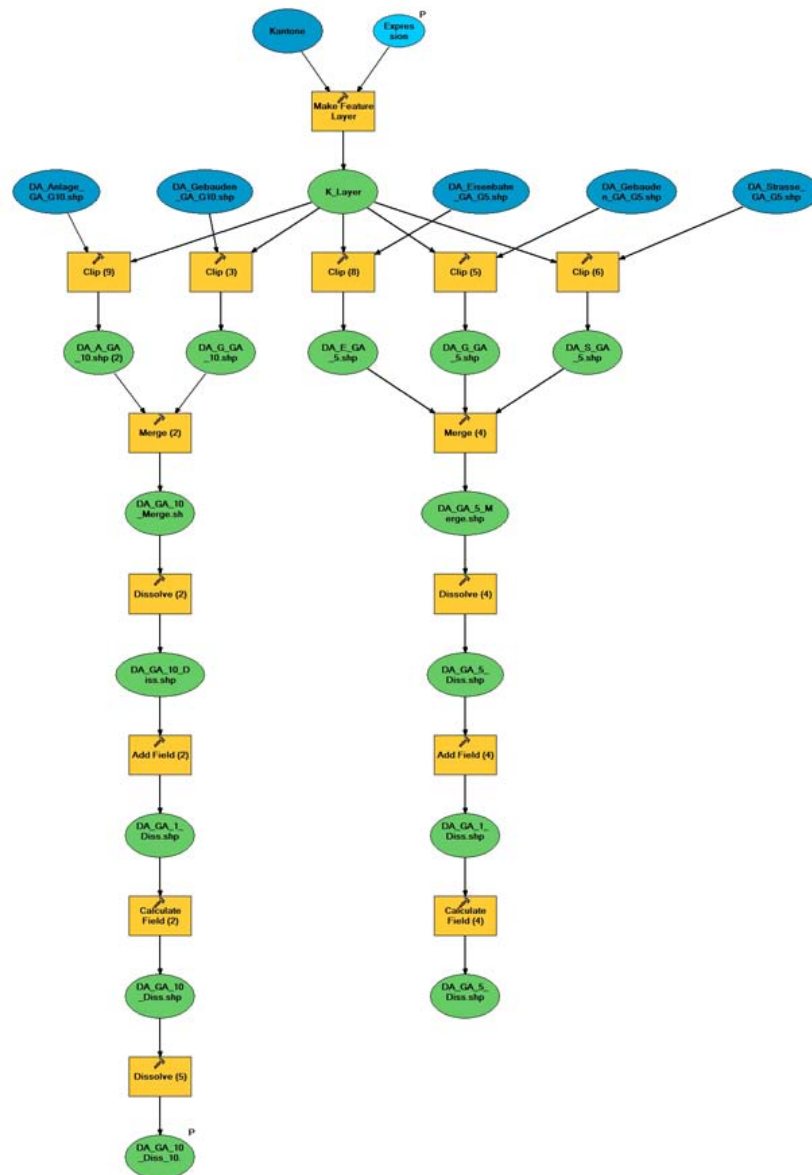
Le résultat de ce géotraitement ne contient aucune superposition de polygones. Si plusieurs surfaces, ayant des pondérations différentes, se superposent, alors seule la surface avec la pondération la plus grande sera conservée, les autres seront supprimées. La Figure 25 suivante montre les différentes pondérations possibles.



Modèle 24: Enjeux utilisés pour le calcul de l'indice du potentiel de dommages (partie 1).

### Détails du modèle

- **Kantone:** Make Feature Layer: **Expression:** (Model Parameter); "KT" = 'AG'
- **Clip:** Input: **DA\_Anlage\_GA\_G3**; Clip: K\_Layer
- **Clip (2):** Input: **DA\_Gebäude\_GA\_G3**; Clip: K\_Layer
- **Clip (4):** Input: **DA\_Eisenbahn\_GA\_G3**; Clip: K\_Layer
- **Clip (7):** Input: **DA\_Strasse\_GA\_G3**; Clip: K\_Layer
- **Merge (3):** DA\_A\_GA\_3, DA\_E\_GA\_3, DA\_G\_GA\_3 und DA\_S\_GA\_3
- **Dissolve (3):** Create Multipart Feature: non checked
- **Add Field (3):** Field Name: GEW, Field Type: SHORT
- **Calculate Field (3):** Field Name: GEW, Expression: 3
- **Clip (10):** Input: **DA\_Eisenbahn\_GA\_G1**; Clip: K\_Layer
- **Clip (11):** Input: **DA\_Anlage\_GA\_G1**; Clip: K\_Layer
- **Merge:** DA\_A\_GA\_1 et DA\_E\_GA\_1
- **Dissolve:** Create Multipart Feature: non checked
- **Add Field:** Field Name: GEW, Field Type: SHORT
- **Calculate Field:** Field Name: GEW, Expression: 1

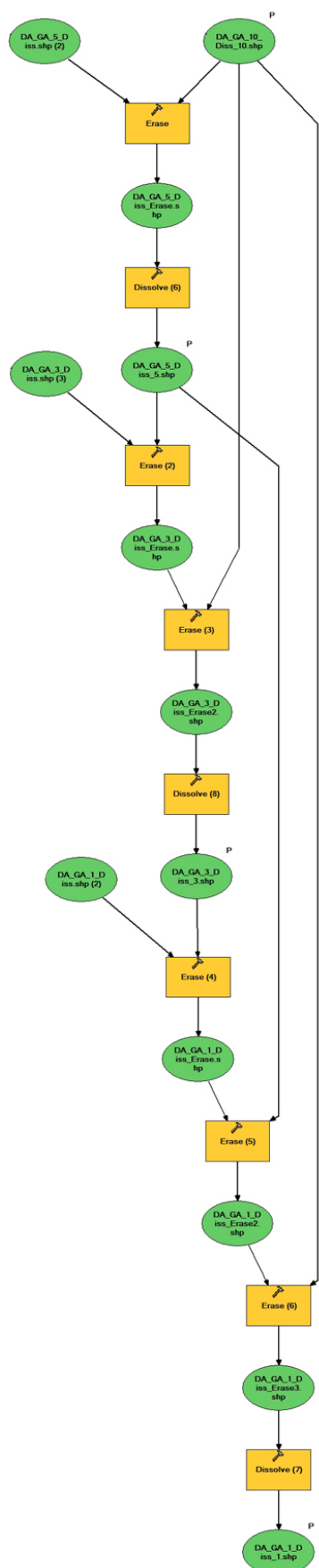


Modèle 25: Enjeux utilisés pour le calcul de l'indice du potentiel de dommages (partie 2).

### Détails du modèle

- **Kantone**: Make Feature Layer: **Expression**: (Model Parameter); "KT" = 'AG'
- **Clip (3)**: Input: **DA\_Gebäude\_GA\_G10**; Clip: K\_Layer
- **Clip (9)**: Input: **DA\_Anlage\_GA\_G10**; Clip: K\_Layer
- **Merge (2)**: DA\_A\_GA\_10 et DA\_G\_GA\_10
- **Dissolve (2)**: Create Multipart Feature: non checked
- **Add Field (2)**: Field Name: GEW, Field Type: SHORT
- **Calculate Field (2)**: Field Name: GEW, Expression: 10
- **Dissolve (5)**: Input: DA\_GA\_10\_Diss; Dissolve Field(s): GEW: checked; Create Multipart Feature: checked
- **Clip (5)**: Input: **DA\_Gebäude\_GA\_G5**; Clip: K\_Layer
- **Clip (6)**: Input: **DA\_Strasse\_GA\_G5**; Clip: K\_Layer
- **Clip (8)**: Input: **DA\_Eisenbahn\_GA\_G5**; Clip: K\_Layer
- **Merge (4)**: DA\_E\_GA\_5, DA\_G\_GA\_5 et DA\_S\_GA\_5
- **Dissolve (4)**: Create Multipart Feature: non checked
- **Add Field (4)**: Field Name: GEW, Field Type: SHORT
- **Calculate Field (4)**: Field Name: GEW, Expression: 5





Modèle 26: Enjeux utilisés pour le calcul de l'indice du potentiel de dommages (partie 3).

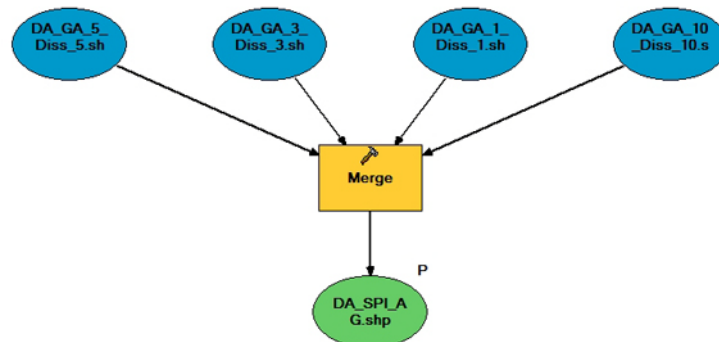
#### Détails du modèle

- **Erase:** Input: DA\_GA\_5\_Diss, Erase: DA\_GA\_10\_Diss
- **Dissolve (6):** Input: DA\_GA\_5\_Erase; Dissolve Field(s): GEW: checked; Create Multipart Feature: checked
- **Erase (2):** Input: DA\_GA\_3\_Diss, Erase: DA\_GA\_5\_Diss
- **Erase (3):** Input: DA\_GA\_3\_Erase, Erase: DA\_GA\_10\_Diss
- **Dissolve (8):** Input: DA\_GA\_3\_Erase\_Erase; Dissolve Field(s): GEW: checked; Create Multipart Feature: checked
- **Erase (4):** Input: DA\_GA\_1\_Diss, Erase: DA\_GA\_3\_Diss
- **Erase (5):** Input: DA\_GA\_1\_Erase, Erase: DA\_GA\_5\_Diss
- **Erase (6):** Input: DA\_GA\_1\_Erase\_Erase, Erase: DA\_GA\_10\_Diss
- **Dissolve (7):** Input: DA\_GA\_1\_Erase\_Erase\_Erase; Dissolve Field(s): GEW: checked; Create Multipart Feature: checked

**Modèle:** DA\_Gew\_Kt\_T2 (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_GA\_10\_Diss, DA\_GA\_5\_Diss, DA\_GA\_3\_Diss et DA\_GA\_1\_Diss.

**Output:** DA\_SPI\_AG.



Modèle 27: Enjeux utilisés pour le calcul de l'indice du potentiel de dommages (partie 4).

#### Détails du modèle

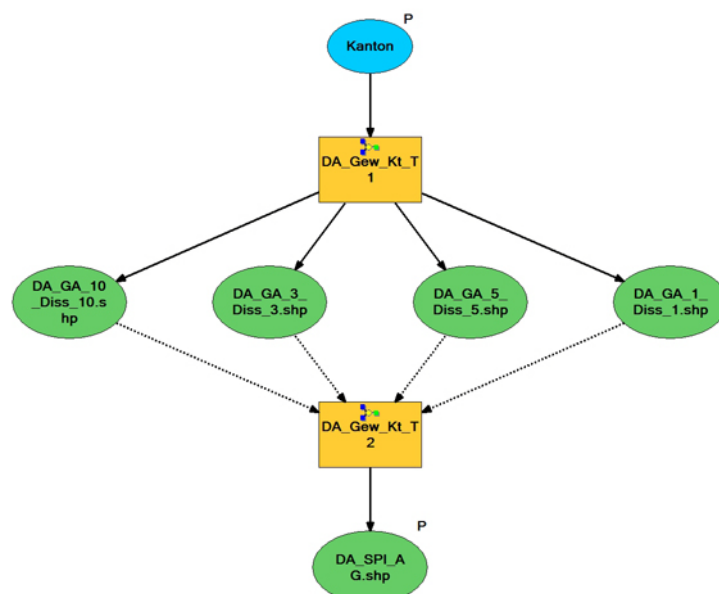
- **Merge:** DA\_GA\_10\_diss, DA\_GA\_5\_diss, DA\_GA\_3\_diss et DA\_GA\_1\_diss

Les deux modèles décrits précédemment (DA\_Gew\_Kt\_T1 et DA\_Gew\_Kt\_T2) peuvent être implémentés dans un seul modèle (DA\_Gew\_Kt).

**Modèle:** DA\_Gew\_Kt (Toolbox: SiPro 93 DAMAGE)

**Input:** DA\_Gew\_Kt\_T1 et DA\_Gew\_Kt\_T2.

**Output:** DA\_SPI\_AG jusqu'à DA\_SPI\_ZH.



Modèle 28: Enjeux utilisés pour le calcul de l'indice du potentiel de dommages (mise ensemble des 2 modèles).

## 5 Processus pertinents (INTERSECT)

### 5.1 Avalanches: zones de déclenchement pertinentes

#### 5.1.1 Description de la modélisation

Comme décrit dans le chapitre 3.1, les avalanches ont été classées en trois catégories: petites, moyennes et grosses. Pour les avalanches petites et moyennes, la zone de déclenchement a été déterminé de la manière suivante: si le périmètre de l'avalanche entre en contact avec l'enjeu, alors la zone de déclenchement correspondante est définie comme pertinente (Figure 27).

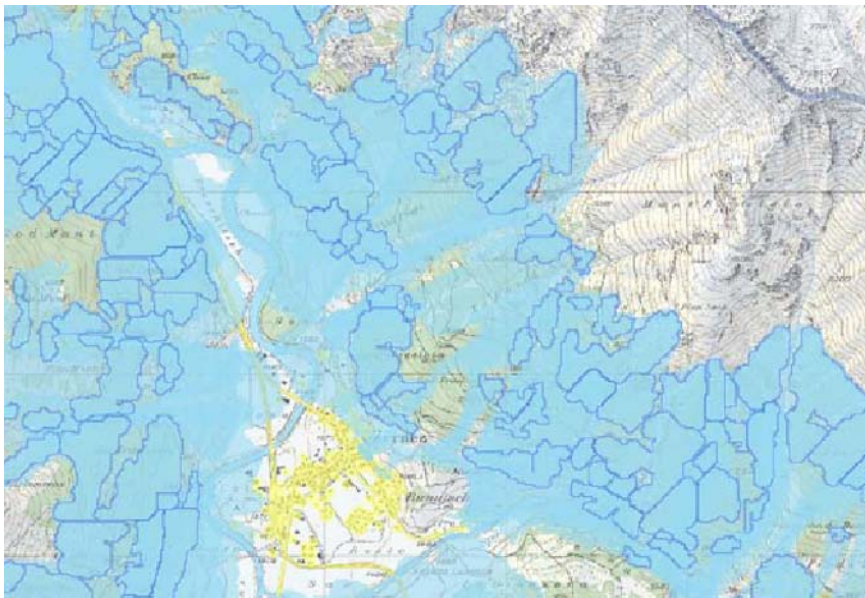


Figure 26: Zones de déclenchement (bleu foncé) pouvant se détacher en forêt et les périmètres d'avalanches correspondants (bleu clair). Les enjeux sont représentés en jaune.

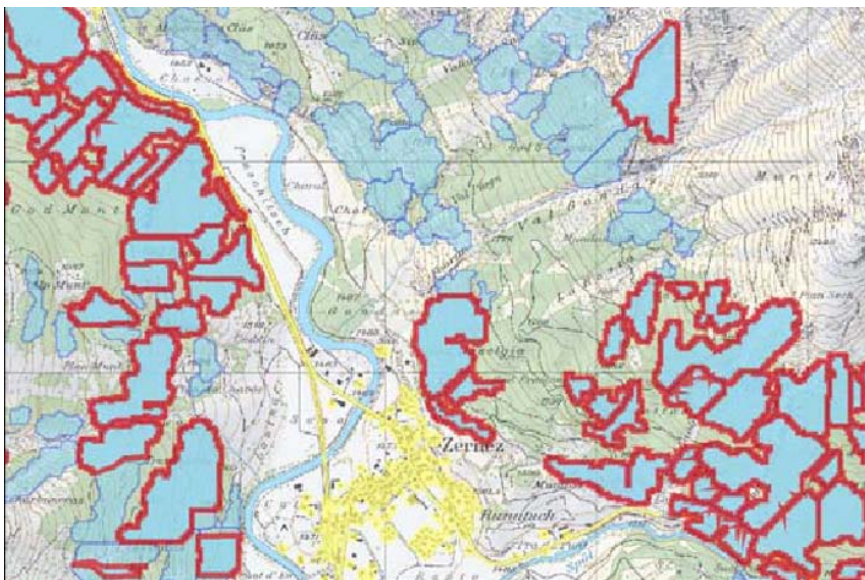


Figure 27: Zones de déclenchement pertinentes (rouge) dont le périmètre peut atteindre un enjeu.



Pour les grosses avalanches, une autre méthode est utilisée pour la modélisation. Comme le montre la Figure 28, l'avalanche (rose) et sa zone de déclenchement (orange) couvrent plusieurs bassins versant, mais l'enjeu (point noir) ne se trouve que dans un seul bassin versant (bleu); ce qui implique que seules les forêts situées dans la partie de la zone de déclenchement du bassin versant concerné peuvent être considérées comme forêts protectrices (vert).

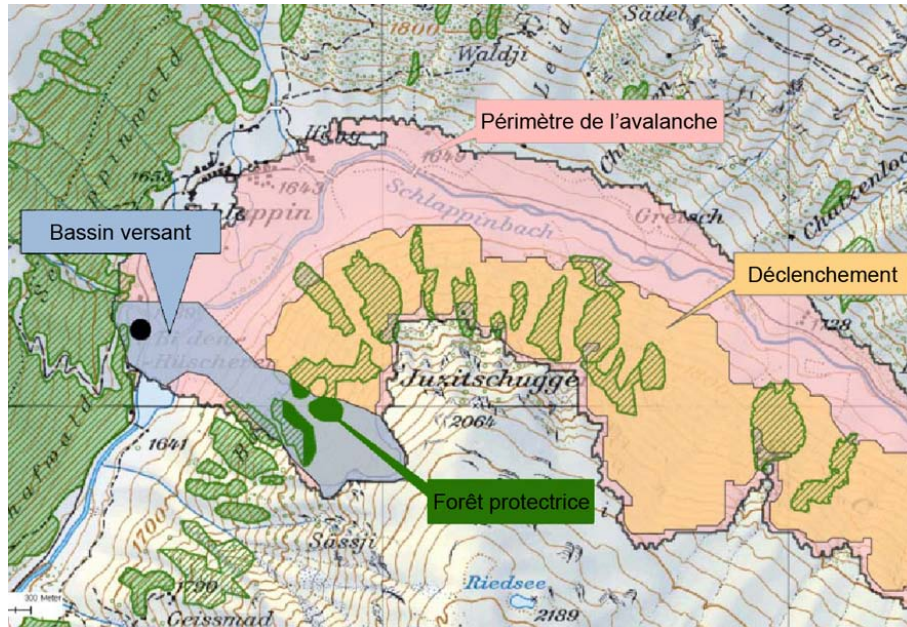


Figure 28: Méthode pour déterminer la zone de déclenchement des grosses avalanches.

De plus, il est possible que l'enjeu se situe en face du bassin versant, et que la force de l'avalanche permette quand même de remonter et d'atteindre cet enjeu (Figure 29).

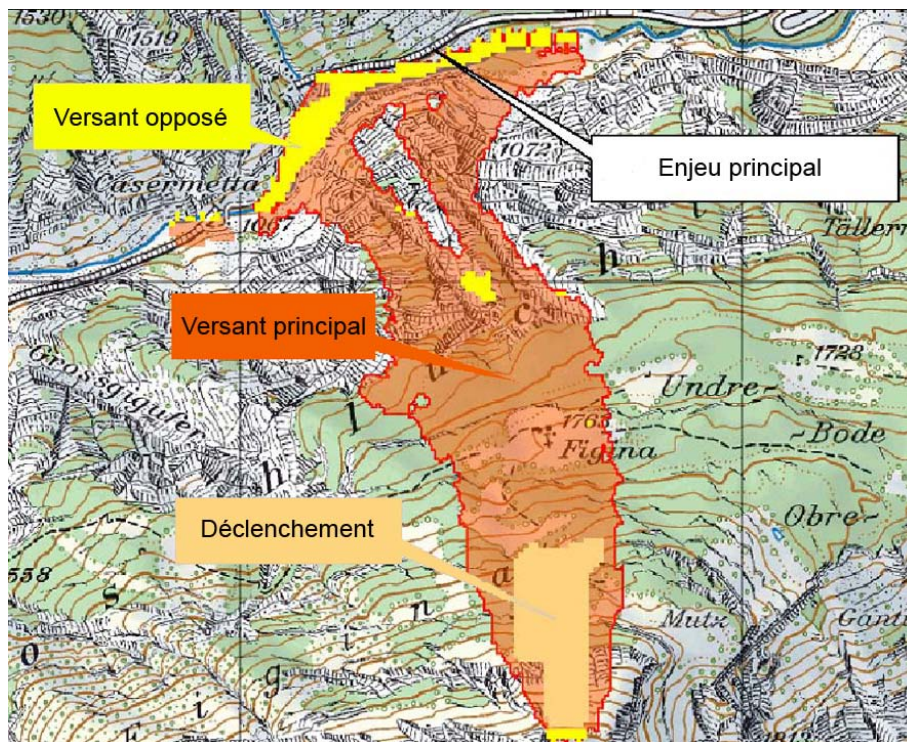


Figure 29: Considération du versant opposé pour déterminer la zone de déclenchement des grosses avalanches.

### 5.1.2 Enjeux selon les régions d'avalanches

**Modèle:** DA\_Lawine\_pro\_Gebiet (Toolbox: SiPro 93 IN Lawine)

**Input:** BA\_Lawine\_Gebiete et DA\_GA

**Output:** 30 Shapes: DA\_adu, DA\_bad ... jusqu'à DA\_wil.

#### Description du modèle

L'enjeux doit être divisé selon les régions d'avalanches (Figure 17). A ces régions a été ajoutée une zone tampon de dix kilomètres afin que tous les enjeux soient bien considérés.

Le modèle a été construit à l'aide d'une liste de valeurs et la Figure 30 montre les critères utilisés:

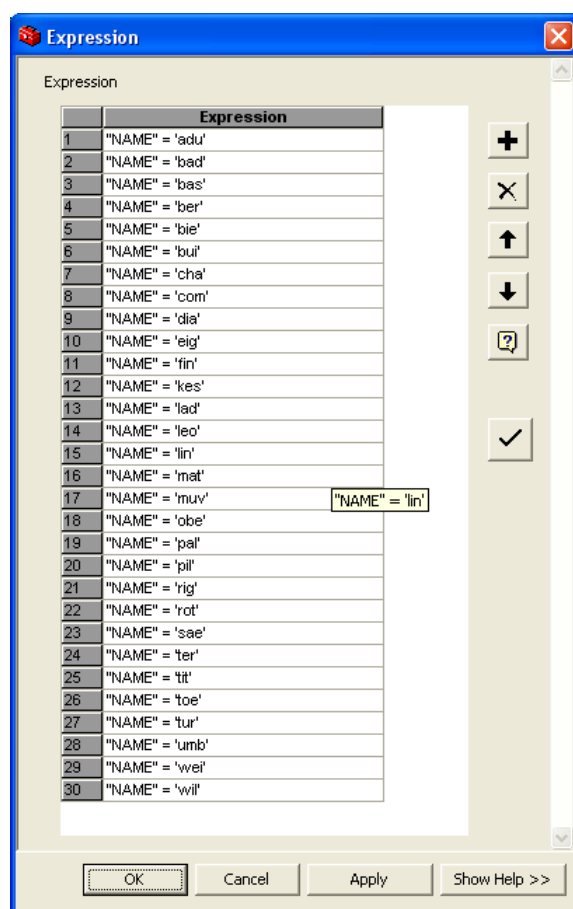
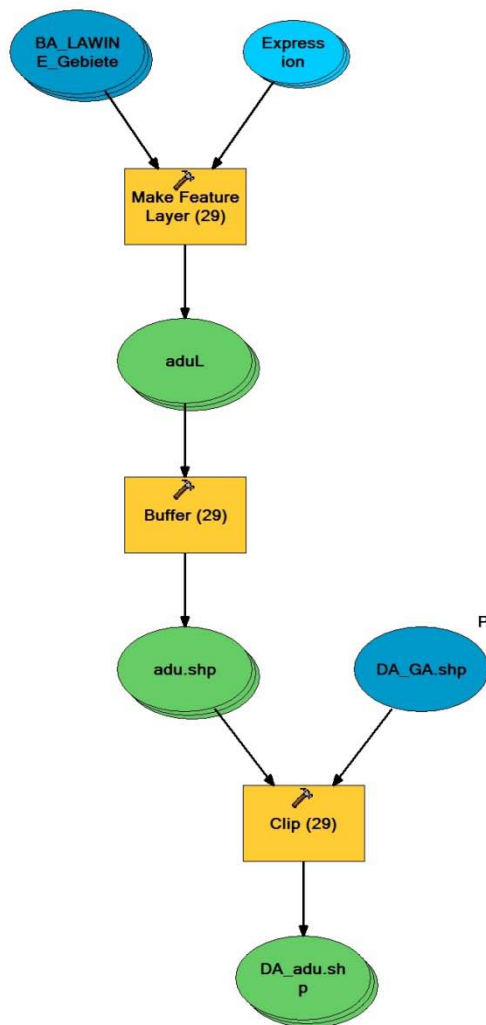


Figure 30: Valeurs de la liste du Modèle 29.



Modèle 29: Enjeux pour les régions d'avalanches.

#### Détails du modèle

- **BA\_LAWINE\_Gebiete:** Make Feature Layer: Expression: voir Figure 30
- **Buffer:** Distance: 10 km, End Type: ROUND
- **DA\_GA:** Make Feature Layer
- **CLIP:** Input Features: DA\_GA, Clip Features: Lawinengebiet

### 5.1.3 Zones de déclenchement pertinentes

L'aléa Avalanche est décrit dans le chapitre 3.1 et les données sont sauvegardées sous D:\SilvaProtect\BASIC\Input. Afin de déterminer les zones de déclenchement pertinentes, les différents scripts ont été programmés à l'aide de Python (version 2.5), voir chapitre 9.3:

- pp\_Launch
- pp\_proc
- pp\_Programm
- pp\_Settings
- pp\_util\_GIS

Les scripts démarrent automatiquement après le lancement du script **pp\_Launch**; seul le script **pp\_Settings** doit être adapté de la manière suivante (Figure 31):

- **src\_Workspace**: endroit où les données à utiliser sont sauvegardées
- **tar\_Folder**: endroit où les données générées sont sauvegardées
- **gebiet**: région d'avalanches traitée
- **DamLayer**: données comprenant les enjeux à partir du modèle **DA\_Lawine\_pro\_Gebiet**
- **dtm**: modèle de terrain utilisé

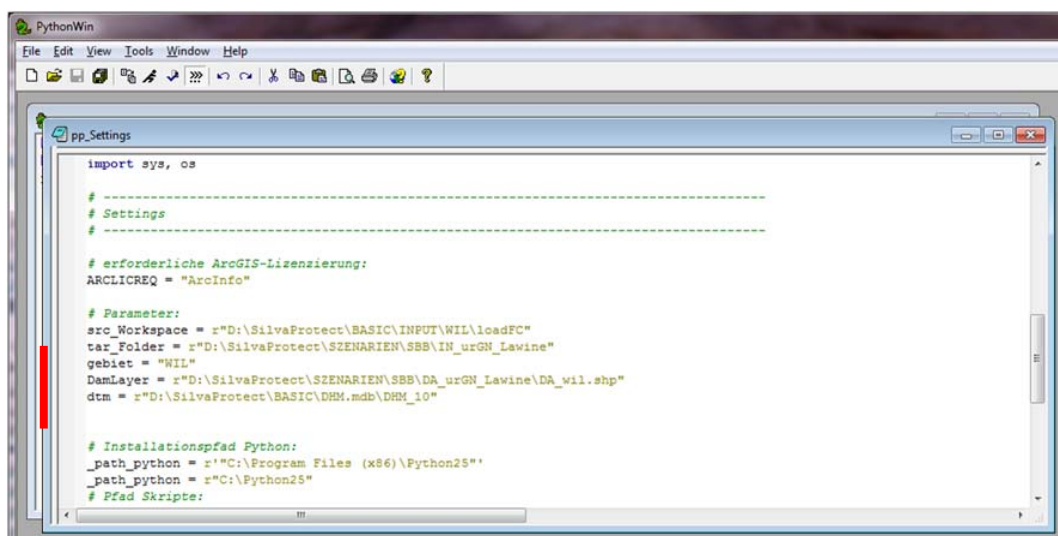


Figure 31: Script pp\_Settings avec les paramètres à introduire.

#### Travaux SIG supplémentaires:

Les résultats obtenus se trouvent dans la base de données créée et sont les suivants **wiln\_rel\_medium**, **wiln\_rel\_small** et **wiln\_sw\_large**.

- **Merge**: Input: **adun\_rel\_small**, **adun\_rel\_medium**, jusqu'à **wiln\_rel\_medium** et **wiln\_rel\_large**
- **Dissolve**: Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "LAW" et la valeur "1":

- **Add Field**: Field Name: LAW, Field Type: SHORT
- **Calculate Field**: Field Name: LAW, Expression: 1
- **Delete Field**: toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape **IN\_LAWINE\_CH**.



## 5.2 Chutes de pierres: trajectoires pertinentes

### 5.2.1 Description de la modélisation

Toutes les trajectoires qui rencontrent un enjeu sont d'abord sélectionnées (Figure 32), puis sont élargies de dix mètres (Figure 33) et finalement "dissoutes" en polygones (Figure 34).

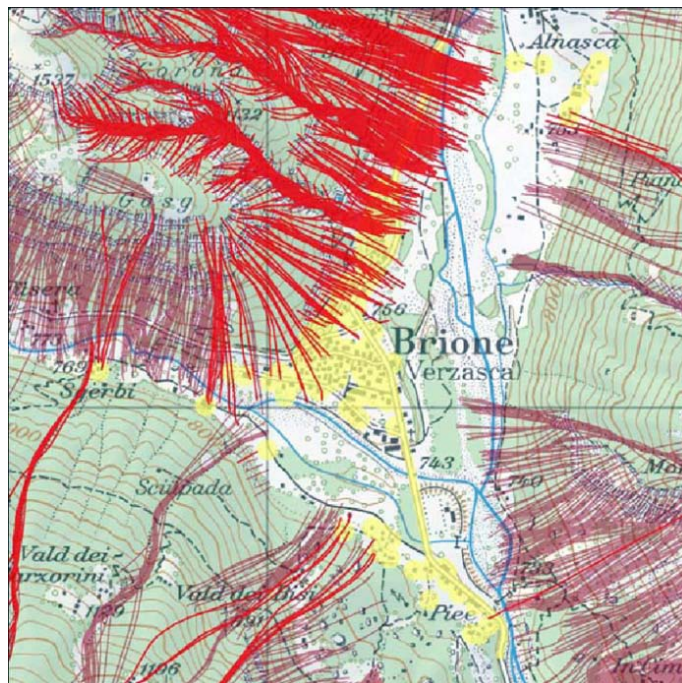


Figure 32: Selection des trajectoires pertinentes (rouge foncé).

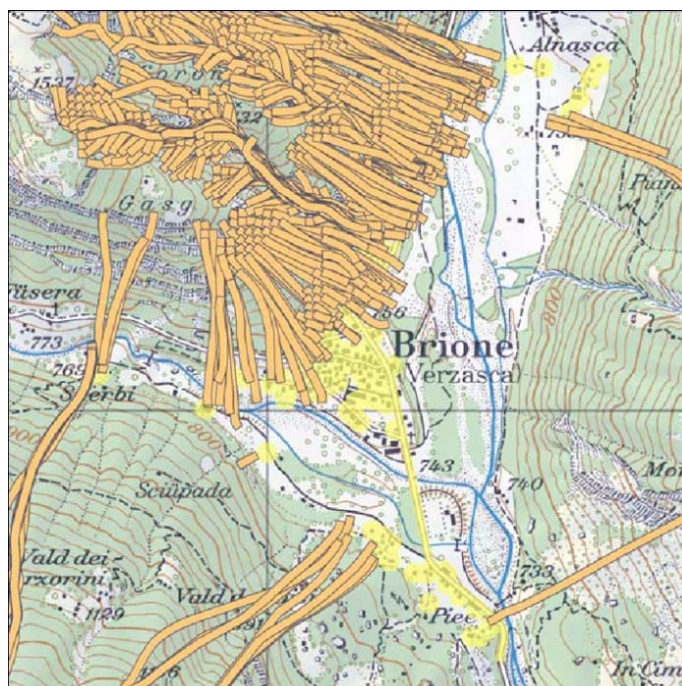


Figure 33: Trajectoires pertinentes élargies de 10 mètres (orange).

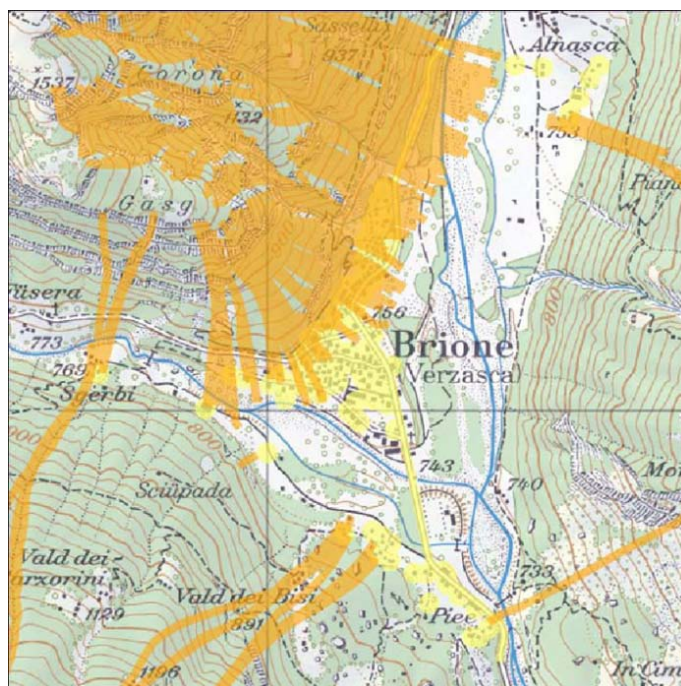


Figure 34: Polygones obtenues à partir de la dissolution des trajectoires.

## 5.2.2 Trajectoires pertinentes pour un carré de 40 km

**Modèle:** IN Sturz07 (Toolbox: SiPro 93 IN STURZ)

**Input:** EV\_Sturz\_07, TILES10 et DA\_GA\_07

**Output:** 4 Shapes: 7bd, 7da, 7db et 7dd.

### Description du modèle

Ce modèle détermine toutes les trajectoires pertinentes du carré 7 selon la Figure 16. Ce modèle peut être utilisé pour tous les types d'enjeux: réseau ferroviaire, installations, bâtiments, etc. Il a été construit à l'aide d'une liste de valeurs et les figures suivantes montrent les critères utilisés.

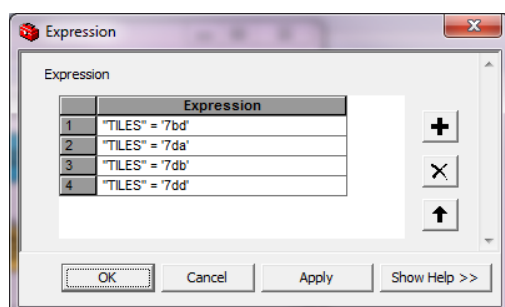


Figure 35: Expression pour le Modèle 30.

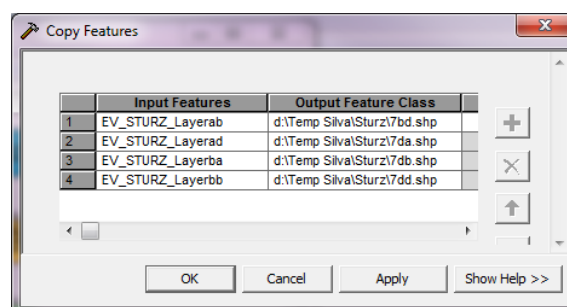
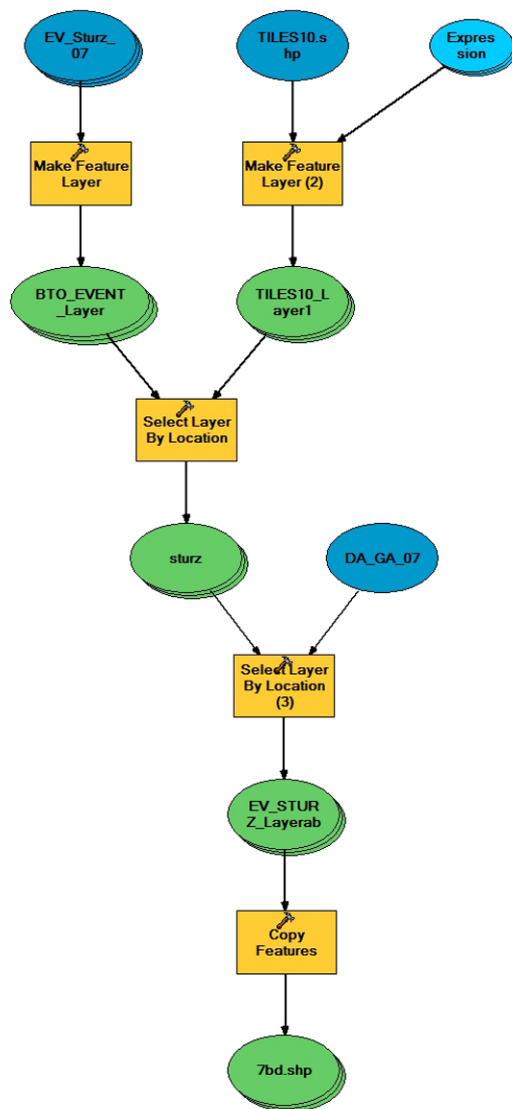


Figure 36: Résultats obtenus avec le Modèle 30.



Modèle 30: Chutes de pierres pertinentes.

#### Détails du modèle

- **EV\_STURZ\_07**: Make Feature Layer, Properties: List of values
- **TILES10**: Make Feature Layer, Expression: (Figure 35)
- **Select Layer By Location**: Input: EV\_STURZ, Selecting: TILES10
- **Select Layer By Location**: Input: Sturz, Selecting: **DA\_GA\_07**, Selection type: SUB-SET\_SELECTION
- **Copy Features**: (Figure 36)

### 5.2.3 Trajectoires pertinentes pour toute la Suisse

**Modèle:** IN STURZ (Toolbox: SiPro 93 IN STURZ)

**Input:** Modèles IN Sturz01, IN Sturz02, ... jusqu'à IN Sturz49  
EV\_STURZ\_01 jusqu'à EV\_STURZ\_49, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49

**Output:** environ 500 Shapes: 1bc, 1bd, jusqu'à 49cc et 49cd.

#### Description du modèle

Pour tous les carrés de la Figure 16, un modèle identique au Modèle 30 a été construit. Tous ces modèles sont ensuite introduits dans un seul et unique modèle. Ainsi, les trajectoires pertinentes de toute la Suisse peuvent être déterminées en une seule fois.



Modèle 31: Chutes de pierres pertinentes pour toute la Suisse.

#### Travaux SIG supplémentaires:

Pour chaque carré obtenu, les trajectoires élargies sont dissoutes (Figure 34) à l'aide du Modèle 1. Tous les Shapes ainsi obtenus sont ensuite mis ensemble.

- **Dissolve:** DR01 jusqu'à DR49.
- **Merge:** Input: 1bc\_diss, 1bd\_diss, jusqu'à 49cc\_diss et 49cd\_diss
- **Dissolve:** Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "Sturz" et la valeur "1":

- **Add Field:** Field Name: STURZ, Field Type: SHORT
- **Calculate Field:** Field Name: STURZ, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Lorsque tous les carrés ont été calculés, alors les Shapes sont mis ensembles (MERGE) et dissous (DISSOLVE). Le résultat final est le Shape IN\_STURZ\_CH.



## 5.3 Glissements de terrain: trajectoires pertinentes

### 5.3.1 Description de la modélisation

Toutes les trajectoires qui rencontrent un enjeu sont d'abord sélectionnées (Figure 37), puis sont élargies de dix mètres (Figure 38) et finalement "dissoutes" en polygones (Figure 39).

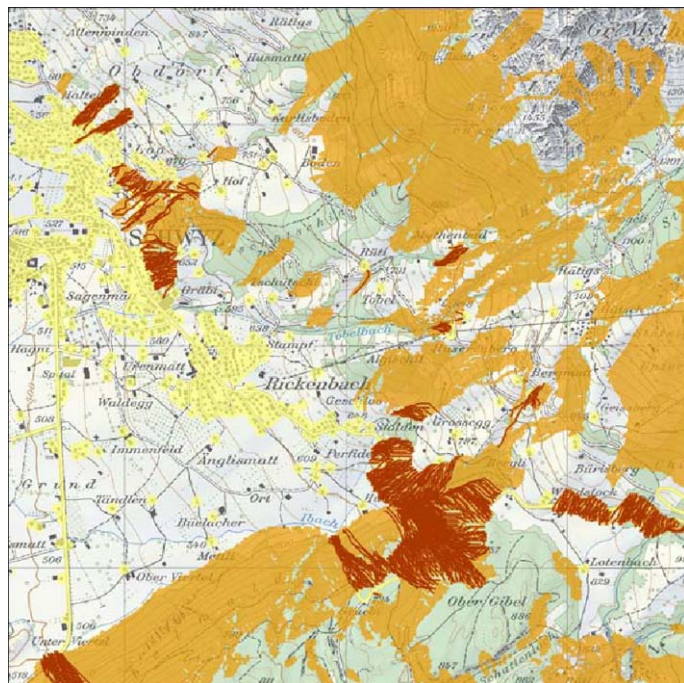


Figure 37: Selection des trajectoires pertinentes (rouge foncé).

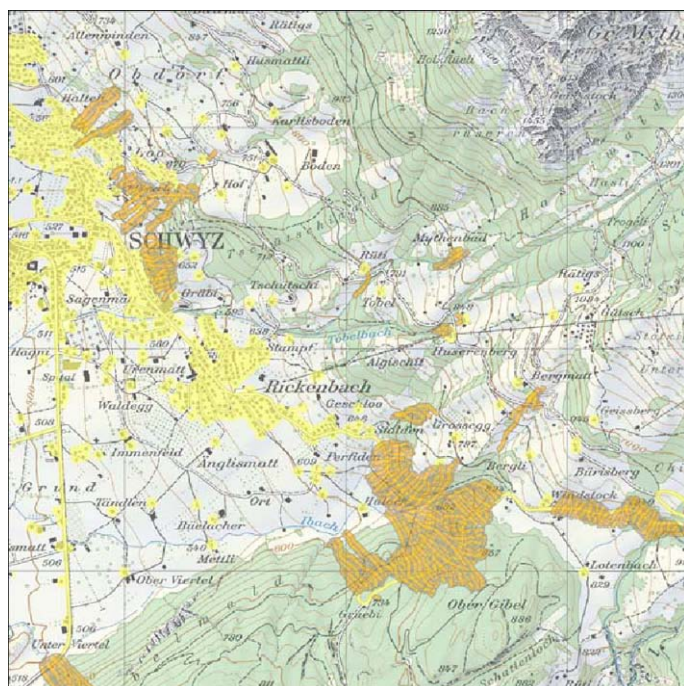


Figure 38: Trajectoires pertinentes élargies de 10 mètres (orange).

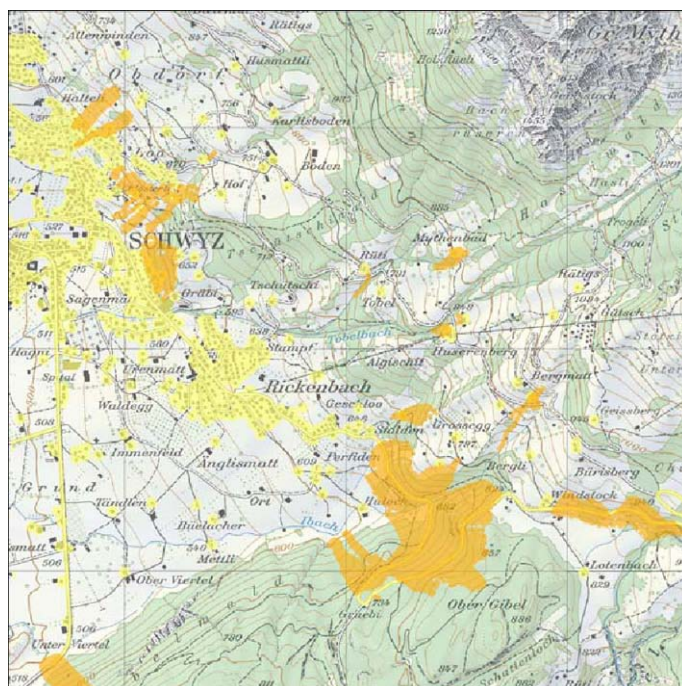


Figure 39: Polygones obtenues à partir de la dissolution des trajectoires.

### 5.3.2 Trajectoires pertinentes pour un carré de 40 km

**Modèle:** IN HM01 (Toolbox: SiPro 93 IN HM)

**Input:** EV\_HANGMURE\_01, TILES10 et DA\_GA\_01

**Output:** 30 Shapes: 1bc, 1bd, 1cc, 1cd, 1da, 1db, 1dc et 1dd.

#### Description du modèle

Ce modèle détermine toutes les trajectoires pertinentes du carré 1 selon la Figure 16. Ce modèle peut être utilisé pour tous les types d'enjeux: réseau ferroviaire, installations, bâtiments, etc. Il a été construit à l'aide d'une liste de valeurs et les figures suivantes montrent les critères utilisés.

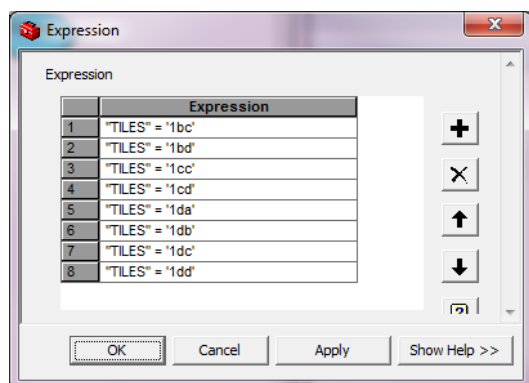


Figure 40: Expression pour le Modèle 32.

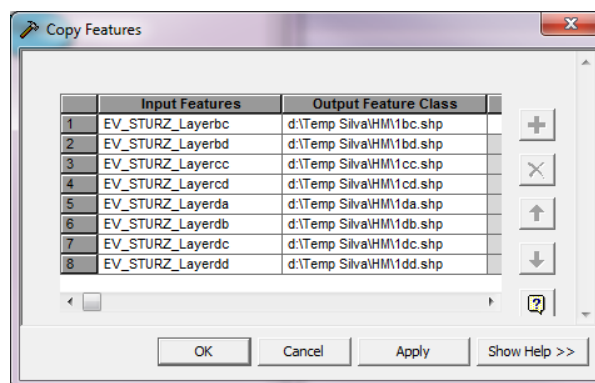
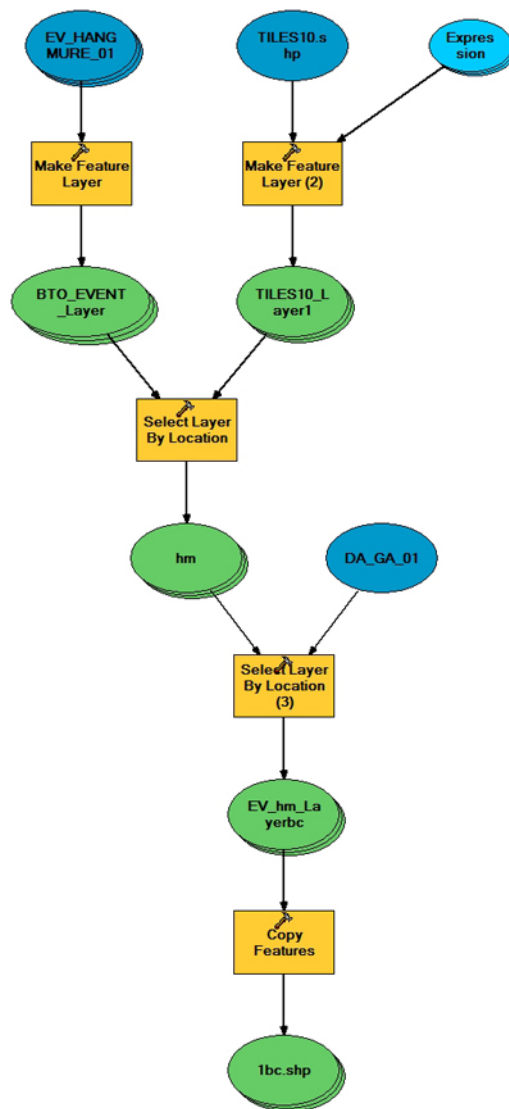


Figure 41: Résultats obtenus avec le Modèle 32.



Modèle 32: Glissements pertinents.

#### Détails du modèle

- **EV\_HANGMURE\_01:** Make Feature Layer, Properties: List of values
- **TILES10:** Make Feature Layer, Expression: (Figure 40)
- **Select Layer By Location:** Input: EV\_HANGMURE, Selecting: TILES10
- **Select Layer By Location:** Input: Sturz, Selecting: DA\_GA\_01, Selection type: SUB-SET\_SELECTION
- **Copy Features:** (Figure 41)



### 5.3.3 Trajectoires pertinentes pour toute la Suisse

**Modèle:** IN HM (Toolbox: SiPro 93 IN HM)

**Input:** Modèles IN HM 01, IN HM 02, ... jusqu'à IN HM 49

EV\_HANGMURE\_01 jusqu'à EV\_HANGMURE\_49, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49

**Output:** environ 500 Shapes: 1bc, 1bd, jusqu'à 49cc et 49cd.

#### Description du modèle

Pour tous les carrés de la Figure 16, un modèle identique au Modèle 32 a été construit. Tous ces modèles sont ensuite introduits dans un seul et unique modèle. Ainsi les trajectoires pertinentes de toute la Suisse peuvent être déterminées en une seule fois.



Modèle 33: Glissements pertinents pour toute la Suisse.

#### Travaux SIG supplémentaires:

Pour chaque carré obtenu, les trajectoires élargies sont dissoutes (Figure 39) à l'aide du Modèle 1. Tous les Shapes ainsi obtenus sont ensuite mis ensemble.

- **Dissolve:** DR01 bis DR49
- **Merge:** Input: 1bc\_diss, 1bd\_diss, jusqu'à 49cc\_diss et 49cd\_diss
- **Dissolve:** Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "HM" et la valeur "1":

- **Add Field:** Field Name: HM, Field Type: SHORT
- **Calculate Field:** Field Name: HM, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Lorsque tous les carrés ont été calculés, alors les Shapes sont mis ensembles (MERGE) et dissous (DISSOLVE). Le résultat final est le Shape IN\_HM\_CH.

## 5.4 Processus liés aux lits pertinents

Comme le montre la Figure 42, la modélisation des processus liés aux cours d'eau est définie à partir du réseau des lits pertinents, c'est-à-dire des ruisseaux pouvant amener du matériel sur un enjeu défini selon le chapitre 4.

Le réseau de lits pertinents est lui défini à partir des processus "débordement avec dépôt" et "lave torrentielle" et des enjeux.

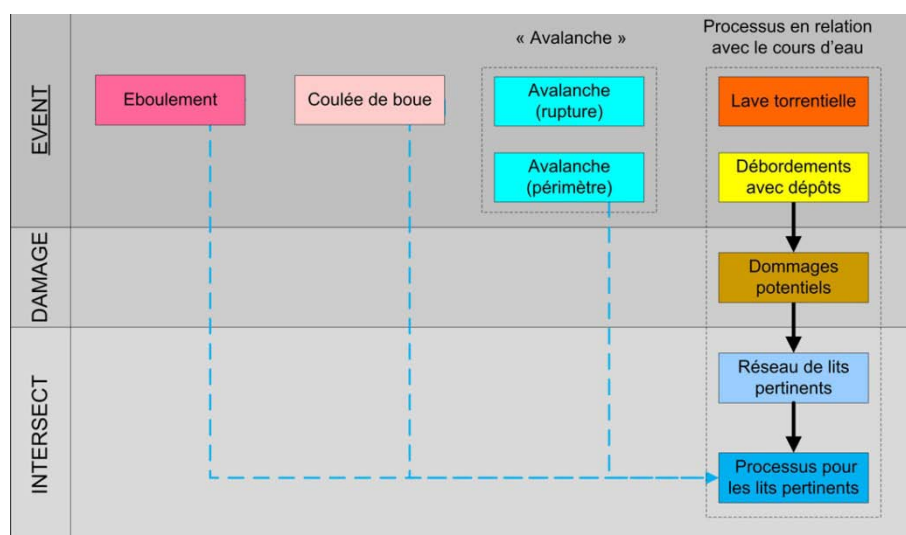


Figure 42: Description de la modélisation des lits pertinents.

Le chapitre 5.4 est donc divisée en deux parties: la première décrit la manière de déterminer le réseau des lits pertinents à l'aide des débordements avec dépôts (5.4.1) et à l'aide des laves torrentielles (5.4.2), la deuxième explique la façon de modéliser les processus liés à ces cours d'eau (5.4.3).

#### 5.4.1.1 Description de la modélisation

This topographic map depicts the Stephan region in the Alps. The central feature is the Ore-see, a large lake surrounded by steep, forested slopes. To the north, the Gandhorn mountain range is visible, with peaks like the Gandhorn and Hohenmald. The map shows a network of rivers and streams, including the Ore-see and the Hohenmald. A yellow highlighted area is present in the central part of the map, likely indicating a specific region of interest. The map also shows various smaller lakes, such as the Stephan-see and the Hohenmald-see. The terrain is characterized by dense contour lines, indicating steep slopes and high elevations. The map is labeled with various geographical features, including mountains, rivers, and lakes.

This topographic map depicts the Lake Mead region, showing the Colorado River and its tributaries. The map includes labels for 'Gandhara', 'Gyashubel', 'H. H. H. H. H.', 'Stephan', 'Ore-ley', and 'A. B. B. B. B.'. The map also shows various roads, rivers, and geographical features.

69/136



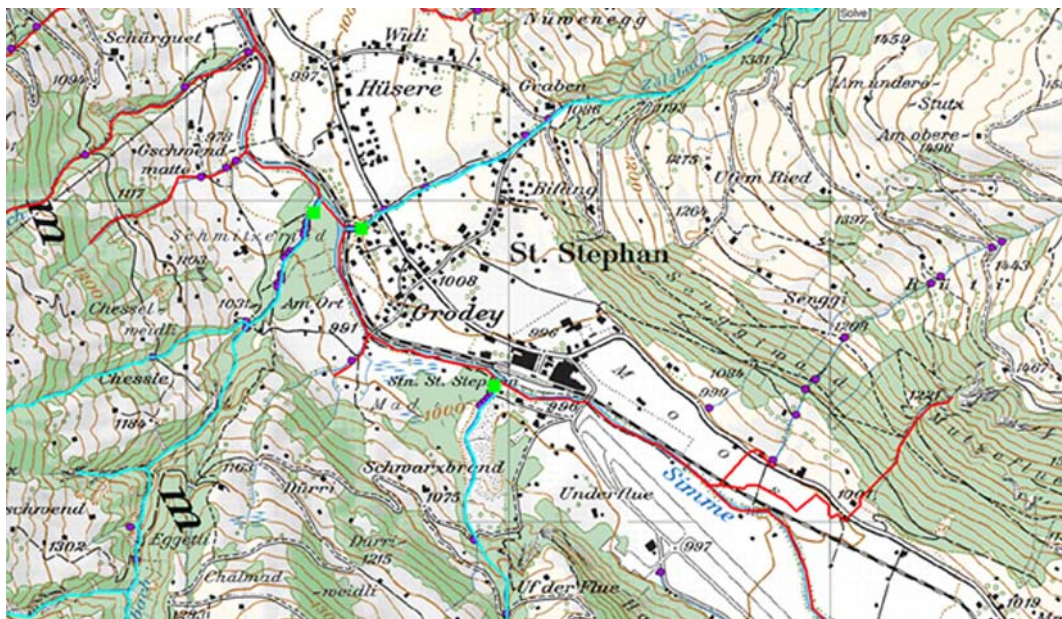


Figure 45: Sélection des lits pertinents à l'aide de l'outil SIG "Utility Network Analyst".

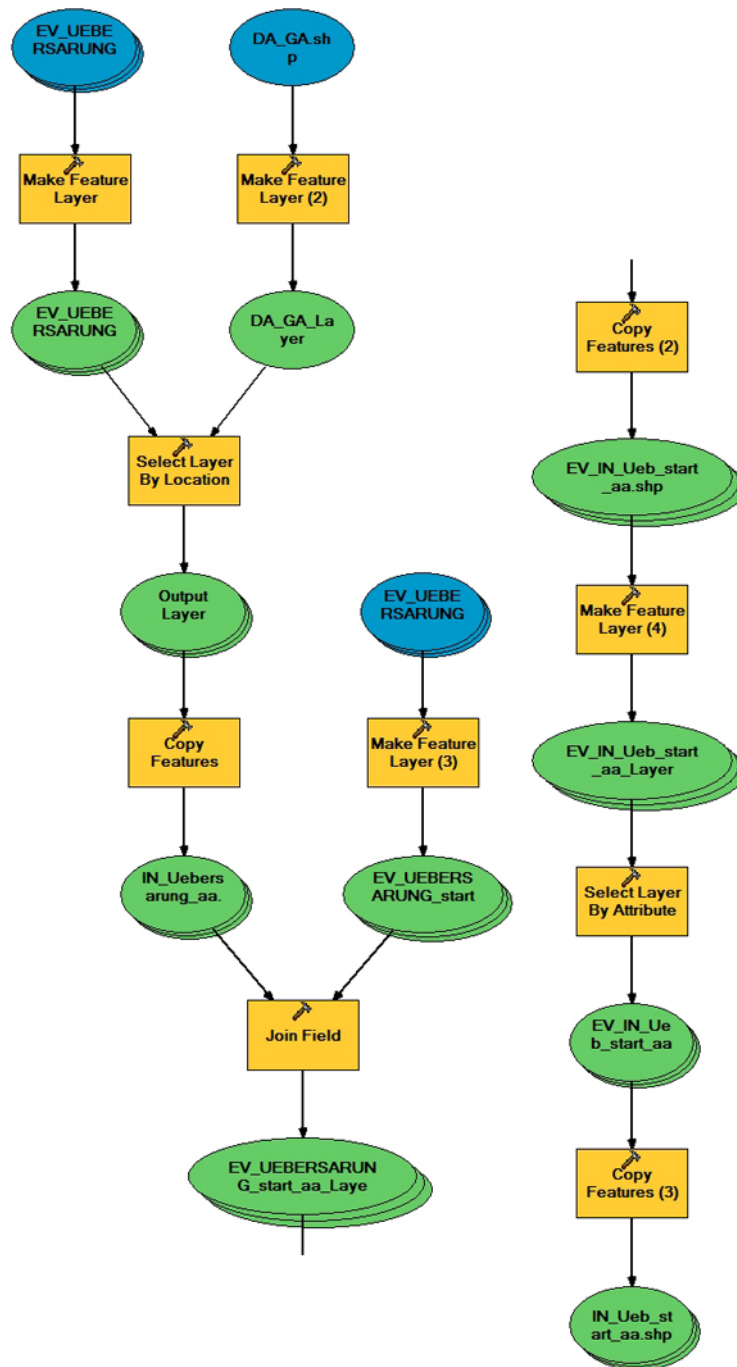
#### 5.4.1.2 Débordements avec dépôts pertinents

Comme le montre la Figure 43, les débordements avec dépôts arrivant sur un enjeu sont d'abord sélectionnés automatiquement selon les régions de débordements avec dépôts (Figure 18).

**Modèle:** IN\_Ueb (Toolbox: SiPro 93 IN\_Ueb)

**Input:** EV\_Uebersarung\_xxx, EV\_Uebersarung\_start\_xxx et DA\_GA.

**Output:** IN\_Ueb\_start\_xxx.



Modèle 34: Débordements avec dépôts pertinents selon les régions.

#### Détails du modèle

- **EV\_Uebersarung\_aa**: Make Feature Layer
- **DA\_GA**: Make Feature Layer
- **Select Layer By Location**: Input: EV\_Uebersarung\_aa\_Layer, Selecting: DA\_GA\_Layer
- **Copy Feature**: Output: IN\_Uebersarung\_aa
- **EV\_Uebersarung\_start\_aa**: Make Feature Layer
- **Join Field**: Input: EV\_Uebersarung\_start\_aa\_Layer, Input Join Field: ID, Join Table: IN\_Uebersarung\_aa, Output Join Field: ID
- **Copy Feature**: Output: EV\_IN\_Ueb\_start\_aa
- **EV\_IN\_Ueb\_start\_aa**: Make Feature Layer
- **Select Layer by Attribute**: ID\_1 > 0
- **Copy Feature**: Output: IN\_Ueb\_start\_aa

### 5.4.1.3 Lits pertinents avec l'outil SIG "Utility Network Analyst"

Cet outil SIG s'utilise de manière semi-automatique. Un premier tri a été fait de manière automatique avec recoupement de couches tandis que la deuxième étape s'est fait à partir d'analyse de cartes.

Le **réseau hydrologique** utilisé ne provient pas du Vector25 de Swisstopo, mais a été construit à partir du modèle numérique de terrain (DTM25); ce qui permet ainsi de construire un réseau et d'indiquer la direction d'écoulement de l'eau (Figure 46).

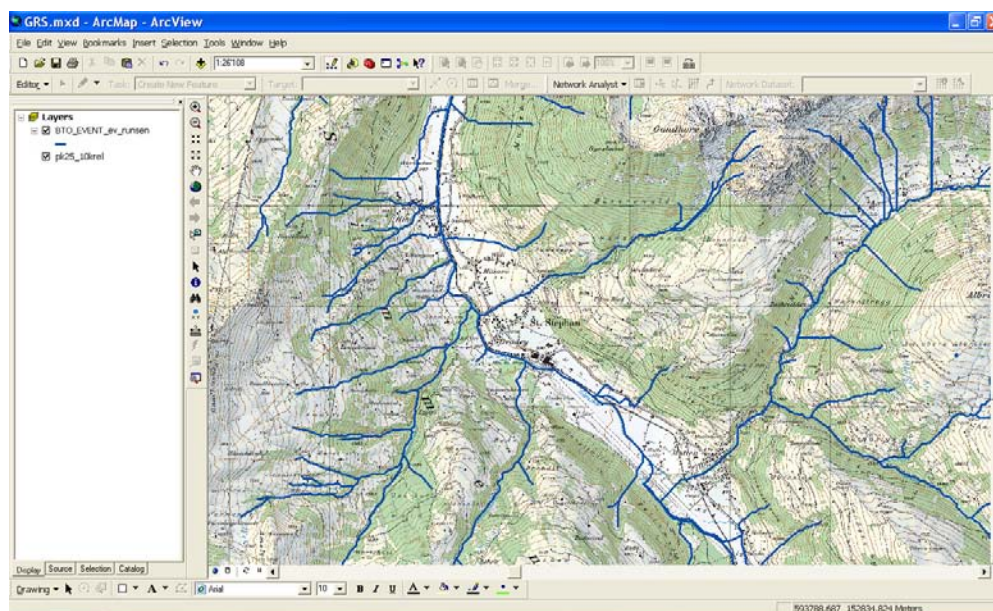


Figure 46: Réseau hydrologique construit intégrant la direction d'écoulement pour l'outil SIG "Utility Network Analyst".

Comme le montre la Figure 47, les endroits où le matériel peut sortir du lit du ruisseau sont marqués en violet (chapitre 5.4.1.2). On peut alors sélectionner ces ruisseaux (points verts) avec l'outil SIG "Utility Network Analyst".

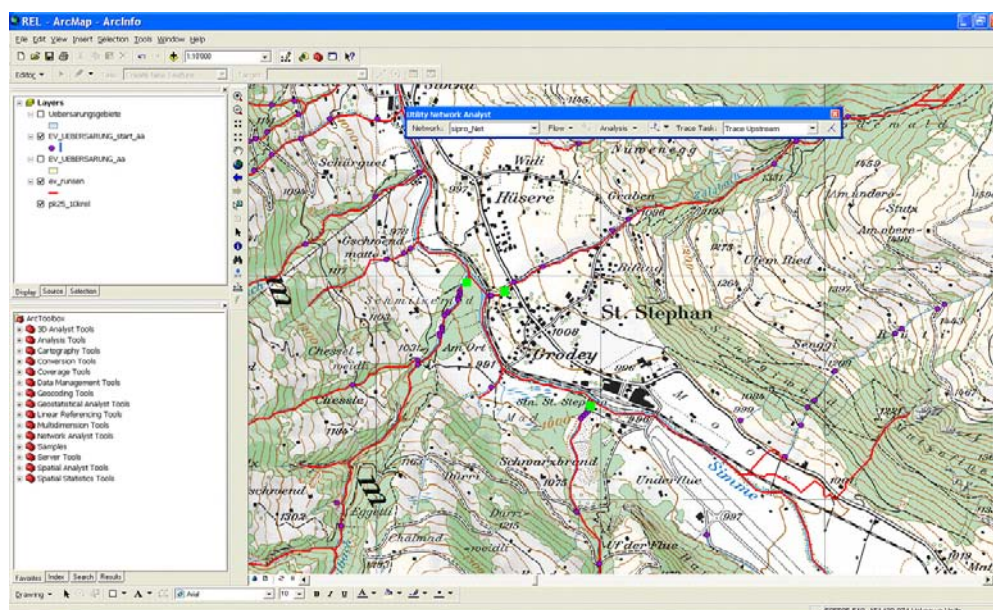


Figure 47: Détermination des lits pertinents.



Avec la fonction "Trace Upstream", la partie amont du ruisseau est automatiquement sélectionnée à partir du point introduit (Figure 48).

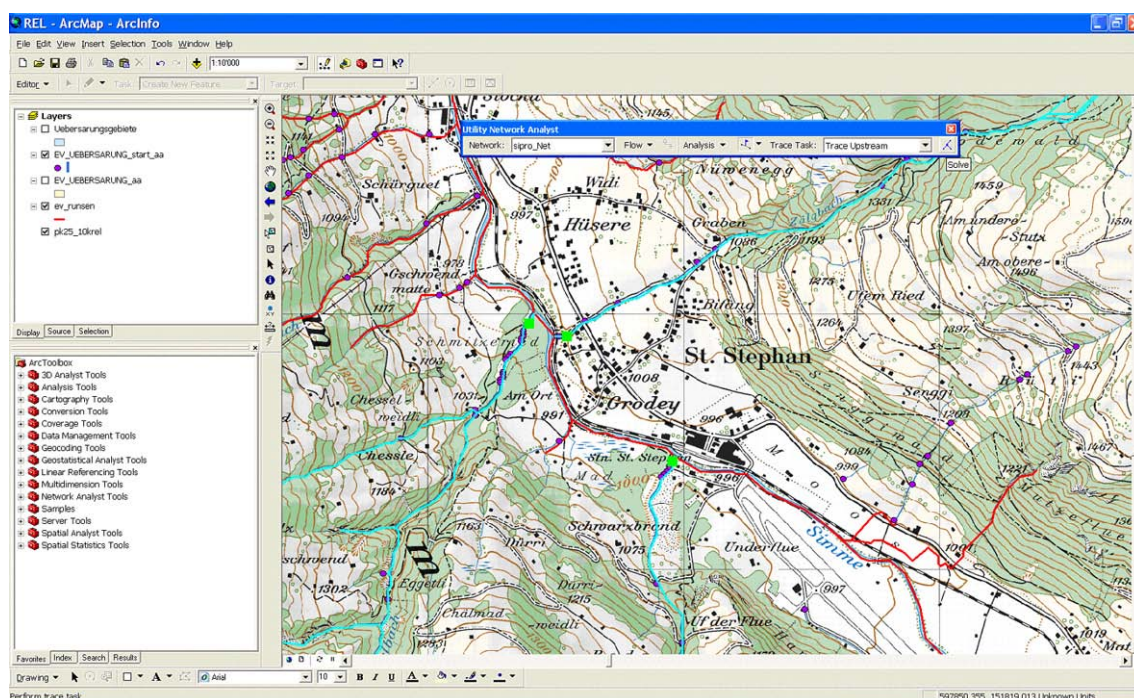


Figure 48: Avec la fonction "Trace Upstream", tout le ruisseau amont est sélectionné.

Pour que cela fonctionne, quelques options doivent être activées (Figure 49). Ainsi l'objet peut être sélectionné et exporté.

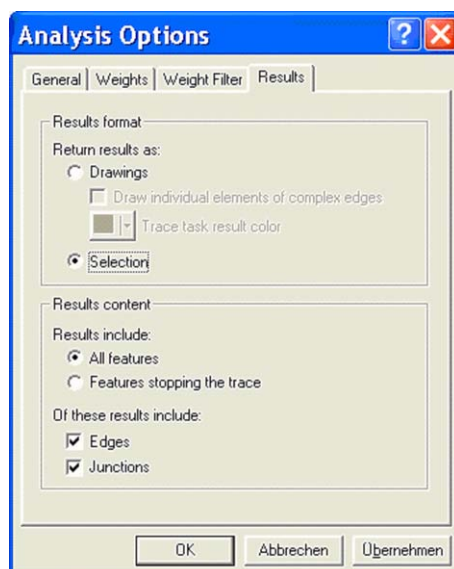


Figure 49: Option de la fonction "Trace Upstream".

Toute la Suisse est ainsi analysée visuellement. Ce travail doit être fait de manière assez conservatrice, sinon de grands bassins versant seront automatiquement sélectionnés. Par exemple si l'on sélectionne le quartier de la Matte à Berne, alors tous les ruisseaux de l'Oberland bernois sont automatiquement sélectionnés!

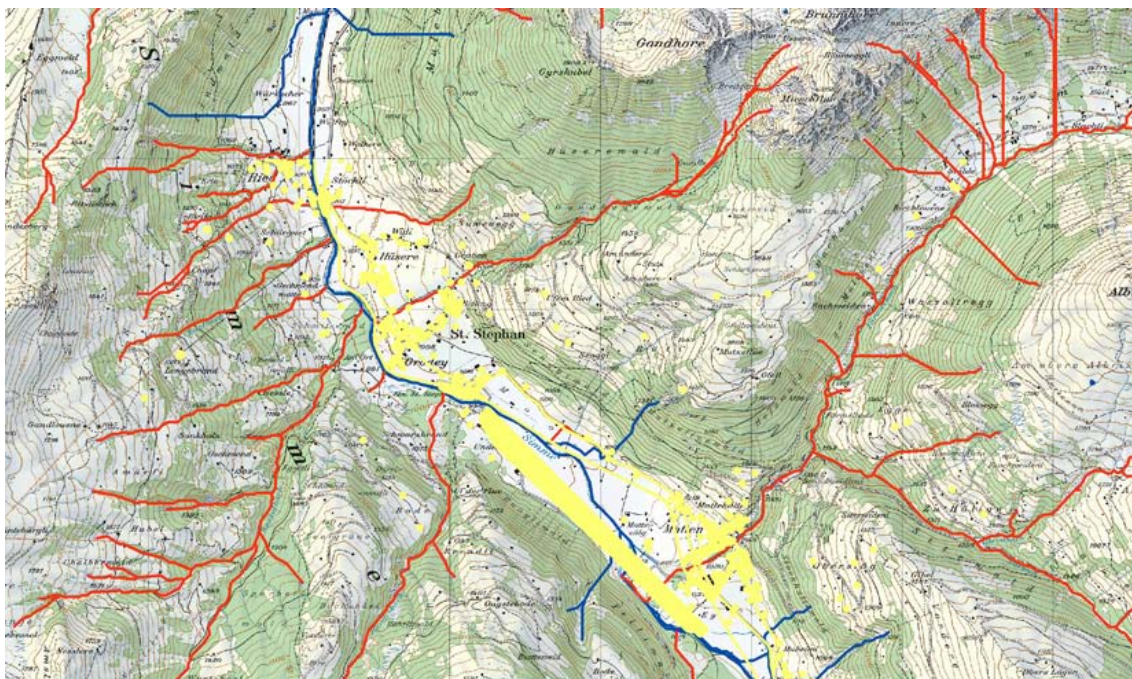


Figure 50: Lits pertinents en rouge.

Le résultat de cette procédure semi-automatisée pour le processus de dangers "débordement avec dépôts" est sauvegardé sous **IN\_RUNSE\_UEB\_CH**.



## 5.4.2 Lits pertinents à l'aide des laves torrentielles

### 5.4.2.1 Description de la modélisation

La modélisation "lave torrentielle" est décrite dans le chapitre 3.4 et est utilisée, en la combinant avec l'enjeux (chapitre 4), pour déterminer les lits pertinents. La première étape est de déterminer les laves torrentielles qui entre en contact avec un enjeux (Figure 51), la deuxième est de déterminer les lits pertinents à partir des laves torrentielles déterminées précédemment (Figure 52).

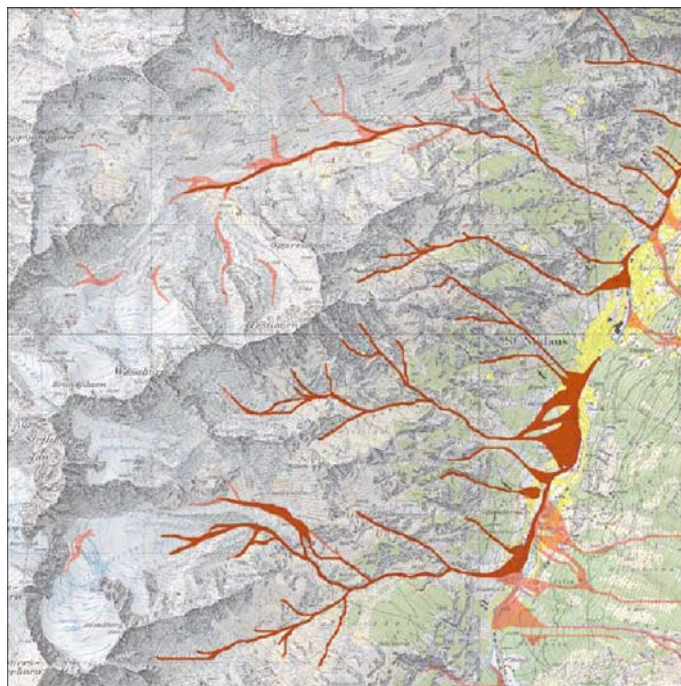


Figure 51: Laves torrentielles pertinentes (rouge foncé).

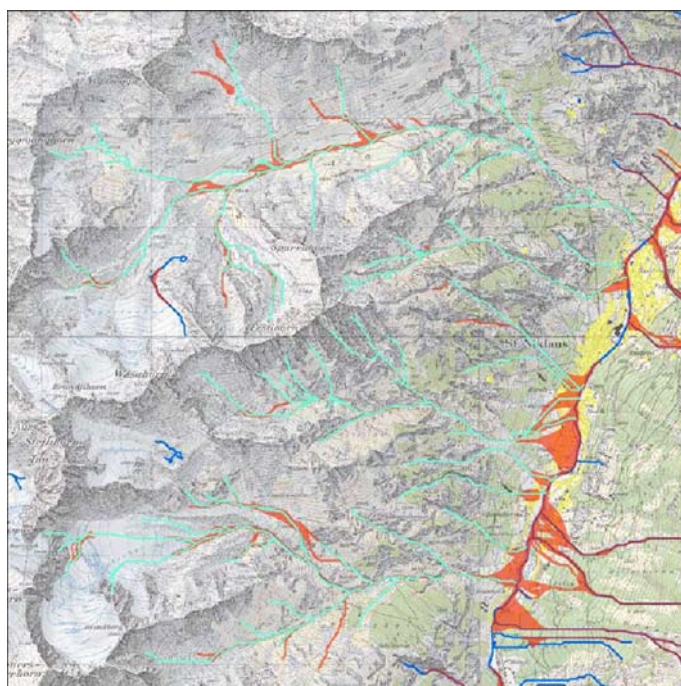


Figure 52: Sélection des lits pertinents (bleu vert) à partir dur réseau hydrologique (bleu).

#### 5.4.2.2 Laves torrentielles pertinentes pour un carré de 40 km

**Modèle:** IN Mur 25 (Toolbox: SiPro 93 IN MUR)

**Input:** EV\_Mur\_25, TILES10 et DA\_GA\_25

**Output:** 15 Shapes: 25ab, 25ac, 25ad, 25ba, 25bb, 25bc, 25bd, 25ca, 25cb, 25cc, 25cd, 25da, 25db, 25dc et 25dd.

#### Description du modèle

Ce modèle détermine toutes les trajectoires pertinentes du carré 25 selon la Figure 16. Ce modèle peut être utilisé pour tous les types d'enjeux: réseau ferroviaire, installations, bâtiments, etc. Il a été construit à l'aide d'une liste de valeurs et les figures suivantes montrent les critères utilisés.

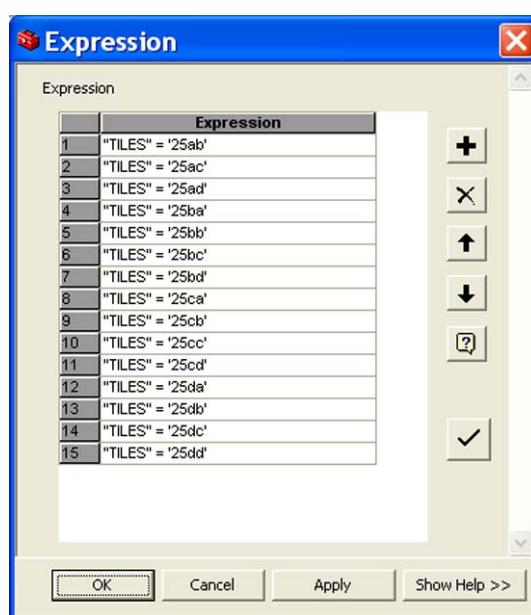


Figure 53: Expression pour le Modèle 35.

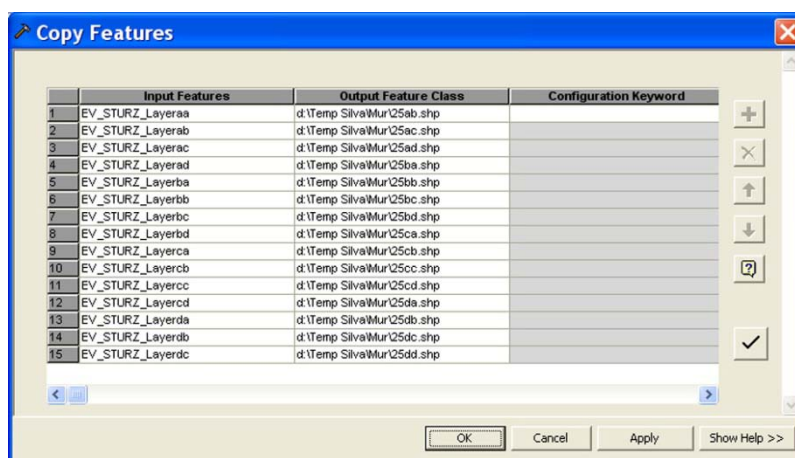
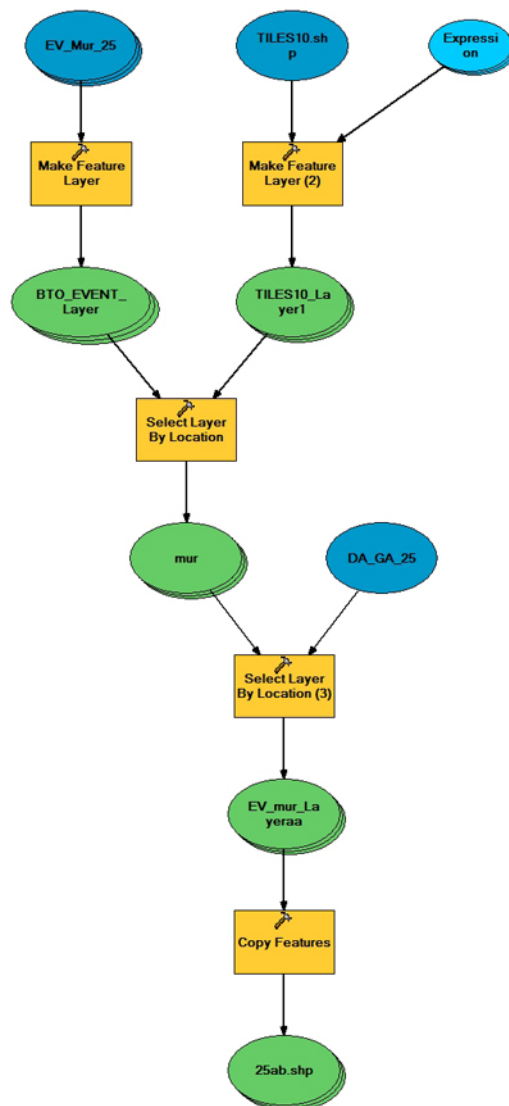


Figure 54: Résultats obtenus avec le Modèle 35.



Modèle 35: Laves torrentielles pertinentes.

#### Détails du modèle

- **EV\_Mur\_25**: Make Feature Layer, Properties: List of values
- **TILES10**: Make Feature Layer, Expression: (Figure 53)
- **Select Layer By Location**: Input: EV\_STURZ, Selecting: TILES10
- **Select Layer By Location**: Input: Sturz, Selecting: **DA\_GA\_07**, Selection type: SUB-SET\_SELECTION
- **Copy Features**: (Figure 54)

### 5.4.2.3 Laves torrentielles pertinentes pour toute la Suisse

**Modèle:** IN Murgang (Toolbox: SiPro 93 IN MUR)

**Input:** Modèles IN Mur 01, IN Mur 02, ... jusqu'à IN Mur 49

EV\_Mur\_01 jusqu'à EV\_Mur\_49, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49

**Output:** environ 500 Shapes: 1bc, 1bd, jusqu'à 49cc et 49cd.

#### Description du modèle

Pour tous les carrés de la Figure 16, un modèle identique au Modèle 35 a été construit. Tous ces modèles sont ensuite introduits dans un seul et unique modèle. Ainsi les trajectoires pertinentes de toute la Suisse peuvent être déterminées en une seule fois.



Modèle 36: Laves torrentielles pertinentes pour toute la Suisse.

#### Travaux SIG supplémentaires:

Pour chaque carré obtenu, les trajectoires élargies sont dissoutes (Figure 51) à l'aide du Modèle 1. Tous les Shapes ainsi obtenus sont ensuite mis ensemble.

- **Dissolve:** DR01 bis DR49
- **Merge:** Input: 1bc\_diss, 1bd\_diss, jusqu'à 49cc\_diss et 49cd\_diss
- **Dissolve:** Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "Mur" et la valeur "1":

- **Add Field:** Field Name: MUR, Field Type: SHORT
- **Calculate Field:** Field Name: MUR, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Lorsque tous les carrés ont été calculés, alors les Shapes sont mis ensemble (MERGE) et dissouts (DISSOLVE). Le résultat final est le Shape IN\_MUR\_CH.



#### 5.4.2.4 Détermination des lits pertinents

Lorsque les surfaces des laves torrentielles ont été déterminées (IN\_MUR\_CH), elles sont superposées au réseau hydrologique utilisé dans le chapitre 5.4.1.3 (Figure 55).

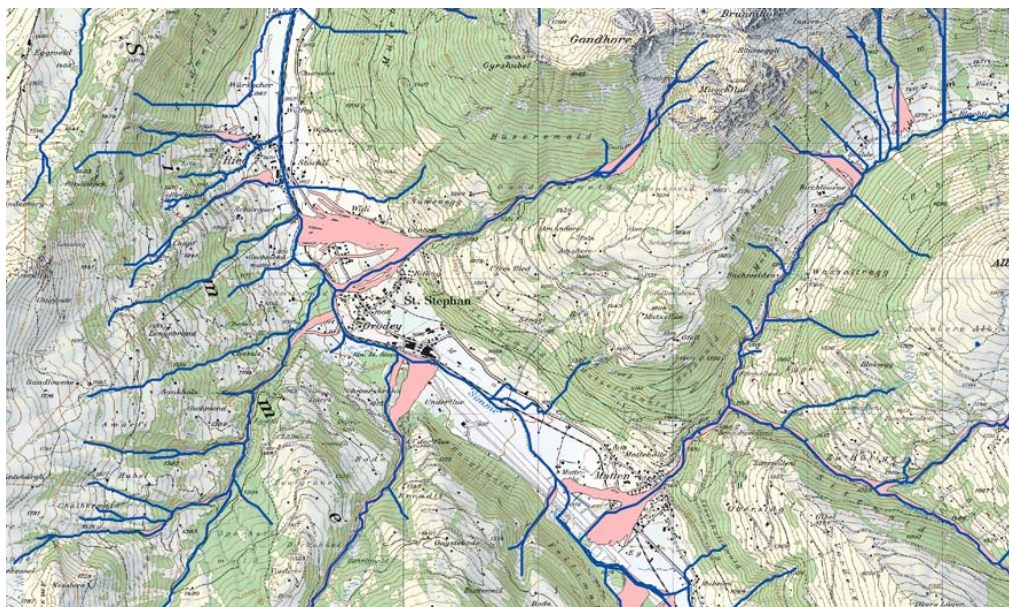


Figure 55: Superposition des laves torrentielles pertinentes (rose) et du réseau hydrologique (bleu).

Afin de déterminer les lits pertinents, l'outil SIG "Utility Network Analyst" est utilisé de la même manière que décrite dans le chapitre 5.4.1.3, mais à partir des laves torrentielles pertinentes (Figure 56) et non pas des points de départ des débordements avec dépôts.

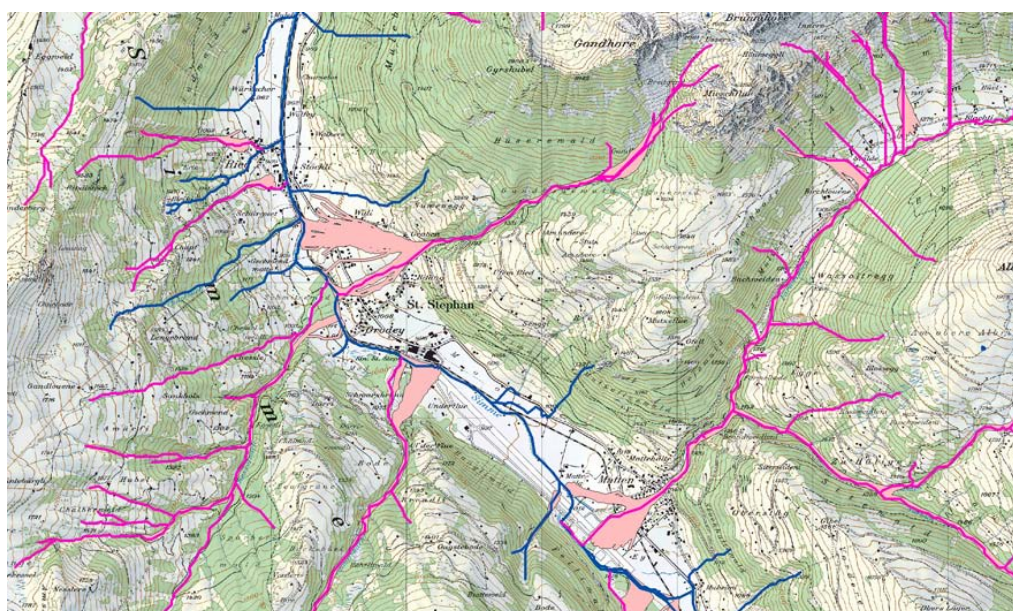


Figure 56: Lits pertinents (bleu) à partir des laves torrentielles pertinentes (rouge).

Le résultat final est le Shape **IN\_RUNSE\_MUR\_CH**.



### 5.4.3 Processus pour les lits pertinents

#### 5.4.3.1 Description de la modélisation

Avant de modéliser les processus liés aux cours d'eau, les deux lits pertinents, obtenus précédemment à partir des processus débordement avec dépôts **IN\_RUNSE\_UEB\_CH** et des laves torrentielles **IN\_RUNSE\_MUR\_CH**, doivent être mis ensemble avec l'outil SIG MERGE. Le résultat obtenu est ainsi sauvé dans le Shape **IN\_rGN\_CH**.

La modélisation des processus liés aux cours d'eau se déroule de la même manière que la modélisation des processus avalanche (chapitre 5.1), chute de pierres (chapitre 5.2) et glissement de terrain (chapitre 5.3). Seul le processus bois flottant est modélisé différemment. C'est pourquoi la description des modélisations décrite dans le chapitre 5.4.3 ne présente que les différences.

#### 5.4.3.2 Avalanches: zones de déclenchement pertinentes

La **première étape** consiste à déterminer les lits pertinents selon les régions d'avalanches comme décrits dans le chapitre 5.1.2. Afin de ne pas créer de modèles supplémentaires, le Shape **IN\_rGN\_CH** va changer de nom et être appelé **DA\_GA**. Ainsi le modèle **DA\_Lawine\_pro\_Gebiet** pourra être utilisé pour les lits pertinents. Le résultat se compose de 30 Shapes (**DA\_adu** jusqu'à **DA\_wil**).

La **deuxième étape** est de définir les zones de déclenchement pertinentes (chapitre 5.1.3).

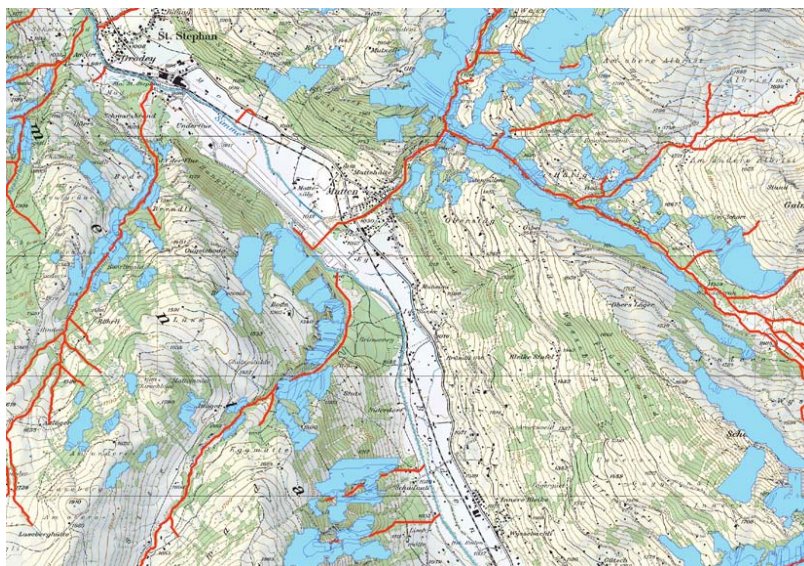


Figure 57: Zones de déclenchement pertinentes (bleu) pour les lits pertinents (rouge).

#### Travaux SIG différents de ceux faits dans le chapitre 5.1.3:

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "GRS\_LAW" et la valeur "1":

- **Add Field:** Field Name: GRS\_LAW, Field Type: SHORT
- **Calculate Field:** Field Name: GRS\_LAW, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape **IN\_GRS\_LAWINE\_CH**.

### 5.4.3.3 Chutes de pierres: trajectoires pertinentes

Le processus est identique à celui décrit dans le chapitre 5.2, mais l'enjeu est différent puisqu'il se compose des lits pertinents déterminés dans le chapitre précédent. Afin de ne pas créer de modèles supplémentaires, le Shape **IN\_rGN\_CH** va changer de nom et être appelé **DA\_GA**. Ainsi les modèles ci-dessous pourront être utilisés.

La modélisation pour chaque carré de 40 km est identique à celle décrite par le modèle **IN\_Sturz07** et celle pour toute la Suisse est identique au modèle **IN STURZ**.

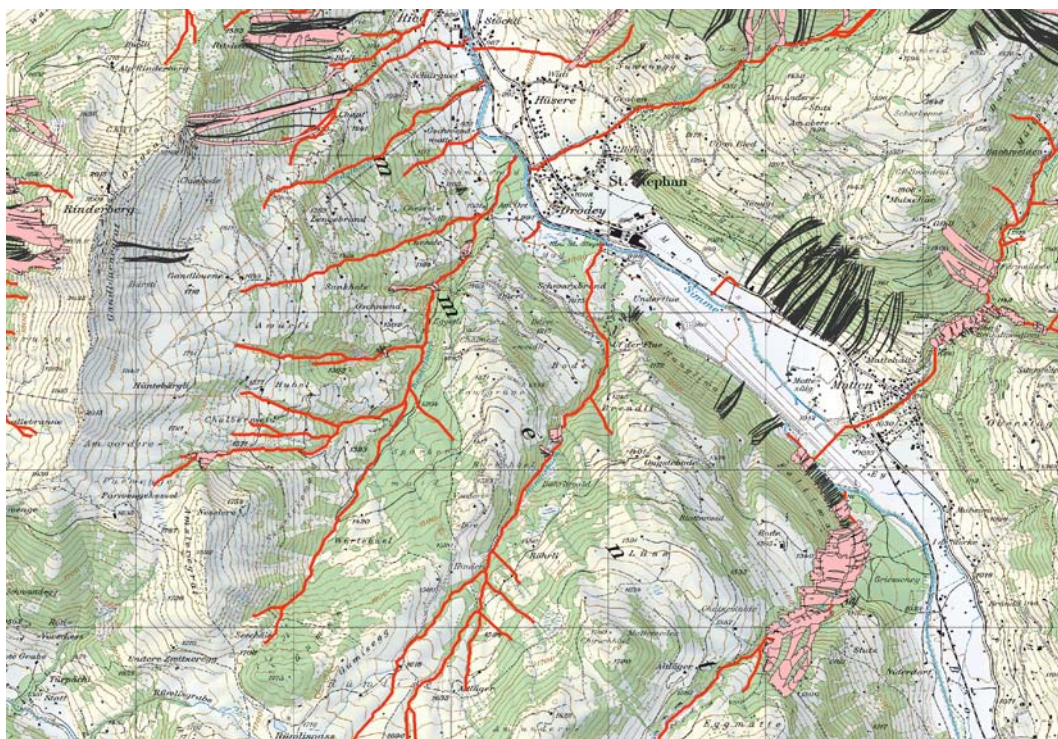


Figure 58: Chutes de pierres pertinentes (rose) pour les lits pertinents (rouge); en noir les autres chutes de pierres.

#### Travaux SIG différents de ceux faits dans le chapitre 5.2.3:

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "GRS\_STURZ" et la valeur "1":

- **Add Field:** Field Name: GRS\_STURZ, Field Type: SHORT
- **Calculate Field:** Field Name: GRS\_STURZ, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape **IN\_GRS\_STURZ\_CH**.



#### 5.4.3.4 Glissements de terrain: trajectoires pertinentes

Le processus est identique à celui décrit dans le chapitre 5.3, mais l'enjeu est différent puisqu'il se compose des lits pertinents déterminés dans le chapitre précédent. Afin de ne pas créer de modèles supplémentaires, le Shape **IN\_rGN\_CH** va changer de nom et être appelé **DA\_GA**. Ainsi les modèles ci-dessous pourront être utilisés.

La modélisation pour chaque carré de 40 km est identique à celle décrite par le modèle **IN HM01** et celle pour toute la Suisse est identique au modèle **IN HM**.

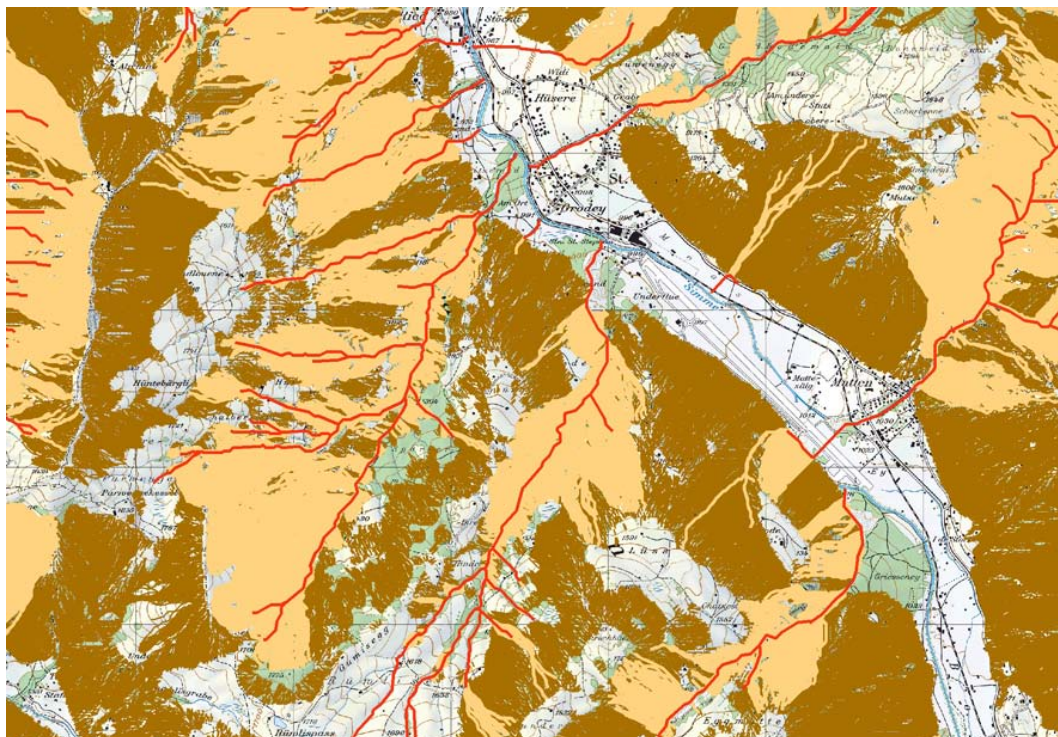


Figure 59: Glissements pertinents (beige) pour les lits pertinents (rouge); en brun les autres glissements.

#### Travaux SIG différents de ceux faits dans le chapitre 5.2.3:

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "GRS\_HM" et la valeur "1":

- **Add Field:** Field Name: GRS\_HM, Field Type: SHORT
- **Calculate Field:** Field Name: GRS\_HM, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape **IN\_GRS\_HM\_CH**.

#### 5.4.3.5 Bois flottants

La mobilisation de bois flottant et l'érosion des berges peuvent être fortement réduites si les forêts sont entretenues les deux côtés des berges.

C'est pourquoi la modélisation du bois flottant comprend toutes les forêts situées à 50 mètres des lits pertinents précédemment définis.

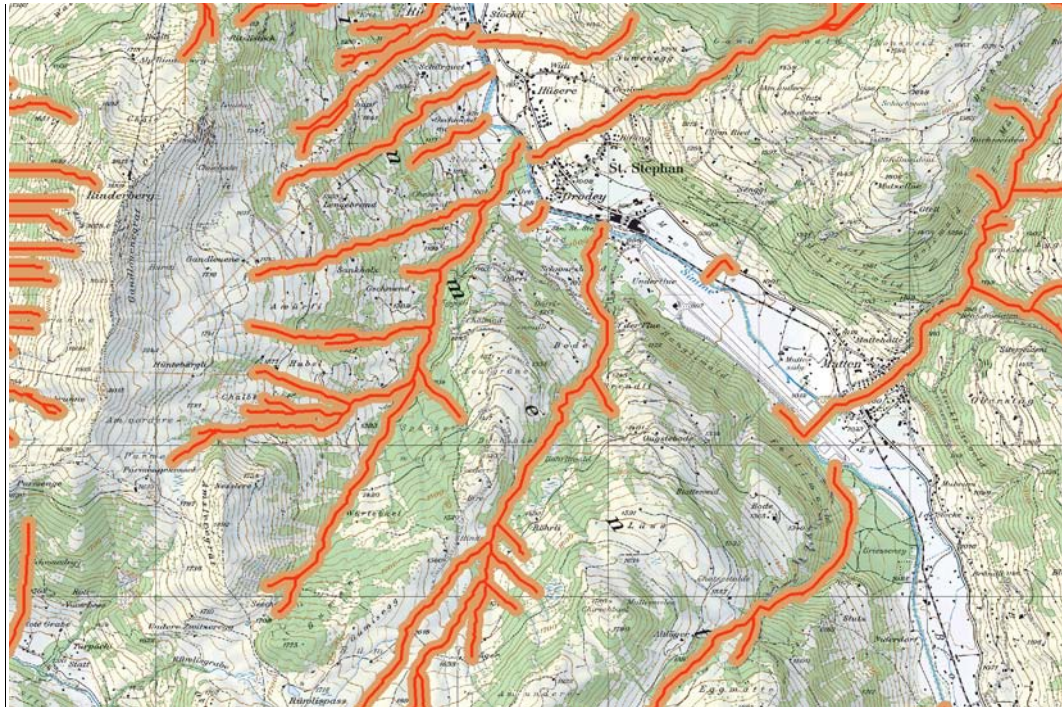


Figure 60: Bois flottants (orange) le long des lits pertinents (rouge).

#### Travaux SIG:

- **Buffer:** Input: `IN_rGN_CH`, Linear Unit 50 Meters
- **Dissolve:** Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "GRS\_SchH" et la valeur "1":

- **Add Field:** Field Name: GRS\_SchH, Field Type: SHORT
- **Calculate Field:** Field Name: GRS\_SchH, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape `IN_GRS_SchHolz_CH`.



### 5.4.3.6 Groupement des processus liés aux lits pertinents

A partir des processus pertinents modélisés pour les lits pertinents (avalanche, chute de pierres, glissement et bois flottant), un Shape unique est créé.

#### Travaux SIG:

- **Union:** Input: [IN\\_GRS\\_LAWINE\\_CH](#), [IN\\_GRS\\_STURZ\\_CH](#), [IN\\_GRS\\_HM\\_CH](#) et [IN\\_GRS\\_SchHolz\\_CH](#).
- **Delete Field:** toutes les colonnes inutiles sont supprimées; seuls les attributs GRS\_LAW, GRS\_STURZ, GRS\_HM et GRS\_SchH sont conservés.

Le résultat obtenu est le Shape **IN\_GRS\_Detail\_CH**, ce qui permet de savoir quels processus sont présents lorsque l'on parle de processus liés aux cours d'eau.

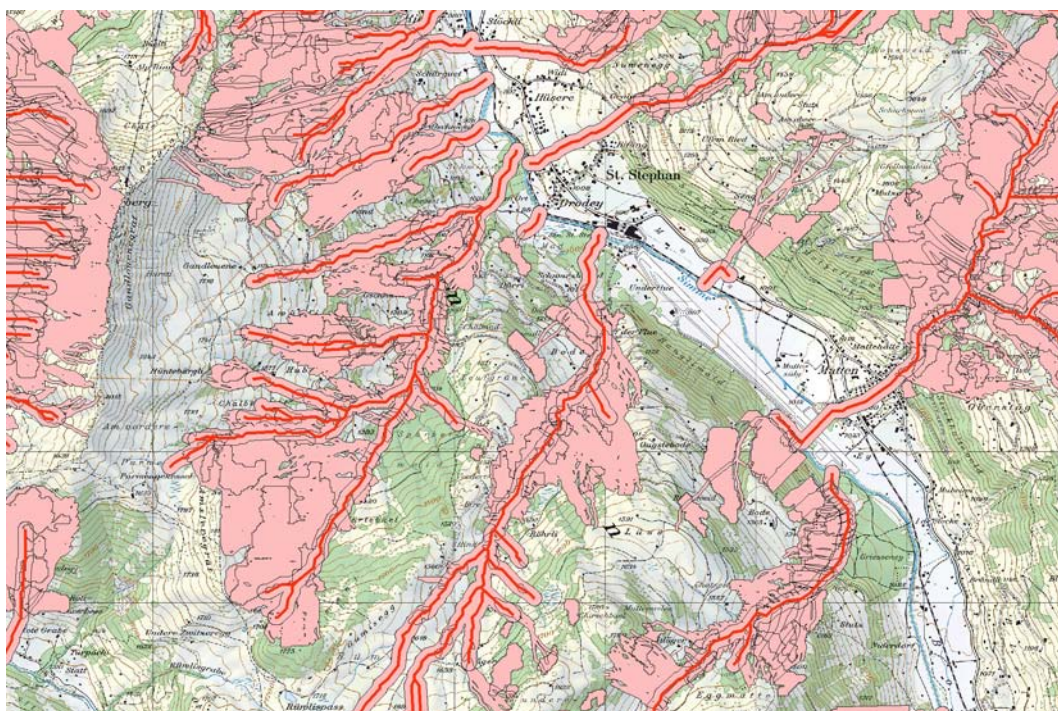


Figure 61: Processus pertinents détaillés (rose) le long des lits pertinents (rouge).

Dans la table de ce Shape, il est ainsi possible de trouver les processus détaillés correspondants:

| Attributes of IN_GRS_Detail_2010 |          |           |          |         |        |
|----------------------------------|----------|-----------|----------|---------|--------|
| FID                              | Shape    | GRS_Sturz | GRS_SchH | GRS_Law | GRS_HM |
| 0                                | Polygon  | 1         | 0        | 0       | 0      |
| 1                                | Polygon  | 1         | 0        | 0       | 0      |
| 2                                | Polygon  | 1         | 0        | 0       | 0      |
| 3                                | Polygon  | 1         | 0        | 0       | 0      |
| 4                                | Polygon  | 1         | 0        | 0       | 0      |
| 5                                | Polygon  | 1         | 0        | 0       | 0      |
| 6                                | Polygon  | 1         | 0        | 0       | 0      |
| 7                                | Polygon  | 1         | 0        | 0       | 0      |
| 8                                | Polygon  | 1         | 0        | 0       | 0      |
| 9                                | Polygon  | 1         | 0        | 0       | 0      |
| 10                               | Polygon  | 1         | 0        | 0       | 0      |
| 11                               | Polygon  | 1         | 0        | 0       | 0      |
| 12                               | Polygon  | 1         | 0        | 0       | 0      |
| 13                               | Polygon  | 1         | 0        | 0       | 0      |
| 14                               | Polygon  | 1         | 0        | 0       | 0      |
| 15                               | Polygon  | 1         | 0        | 0       | 0      |
| 16                               | Polygon  | 1         | 0        | 0       | 0      |
| 17                               | Polygon  | 1         | 0        | 0       | 0      |
| 18                               | Polygon  | 1         | 0        | 0       | 0      |
| 19                               | Polygon  | 1         | 0        | 0       | 0      |
| 20                               | Polygon  | 1         | 0        | 0       | 0      |
| 21                               | Polygon  | 1         | 0        | 0       | 0      |
| 22                               | Polygon  | 1         | 0        | 0       | 0      |
| 23                               | Boundary | 1         | n        | n       | n      |

Figure 62: Processus détaillés des processus liés aux cours d'eau.

Finalement, ce Shape est dissout afin d'avoir une couche propre contenant les processus liés aux cours d'eau, sans distinction détaillée des processus. Ce dernier contient donc seulement l'information s'il y a un processus pertinent lié aux cours d'eau, mais sans connaître exactement ce processus.

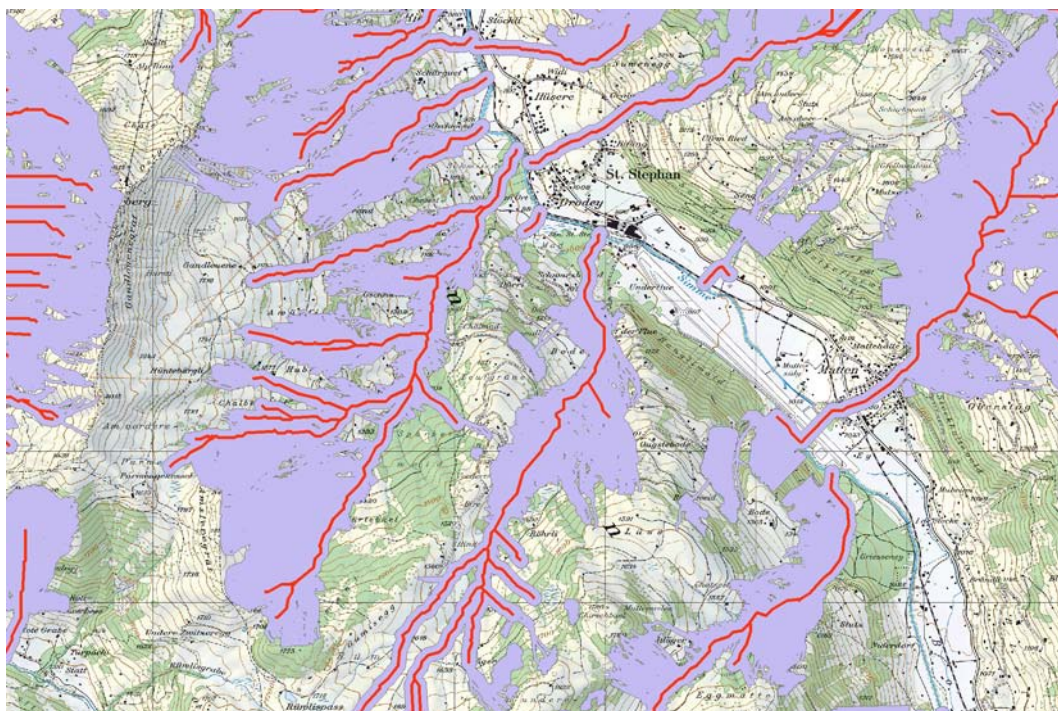


Figure 63: Processus pertinents (violet) le long des cours d'eau pertinents (rouge).

#### Travaux SIG:

- **Dissolve:** Input: [IN\\_GRS\\_Detail\\_CH](#), Create Multipart Feature: non checked

Une nouvelle colonne est introduite dans le tableau des attributs avec le nom "GRS" et la valeur "1":

- **Add Field:** Field Name: GRS, Field Type: SHORT
- **Calculate Field:** Field Name: GRS, Expression: 1
- **Delete Field:** toutes les colonnes inutiles sont supprimées.

Le résultat final est le Shape [IN\\_GRS\\_CH](#)



## 5.5 Groupement des processus pertinents

A partir des processus pertinents modélisés dans les chapitres 5.1 à 5.4 (avalanche, chute de pierres, glissement et processus liés aux cours d'eau), un Shape unique est créé.

### Travaux SIG:

- **Union:** Input: **IN\_LAWINE\_CH**, **IN\_STURZ\_CH**, **IN\_HM\_CH** et **IN\_GRS\_CH**.
- **Delete Field:** toutes les colonnes inutiles sont supprimées; seuls les attributs LAW, STURZ, HM et GRS sont conservés.

Le résultat obtenu est le Shape **IN\_Prozesse\_CH**, ce qui permet de savoir exactement quels processus sont présents lorsque l'on parle de processus pertinents.

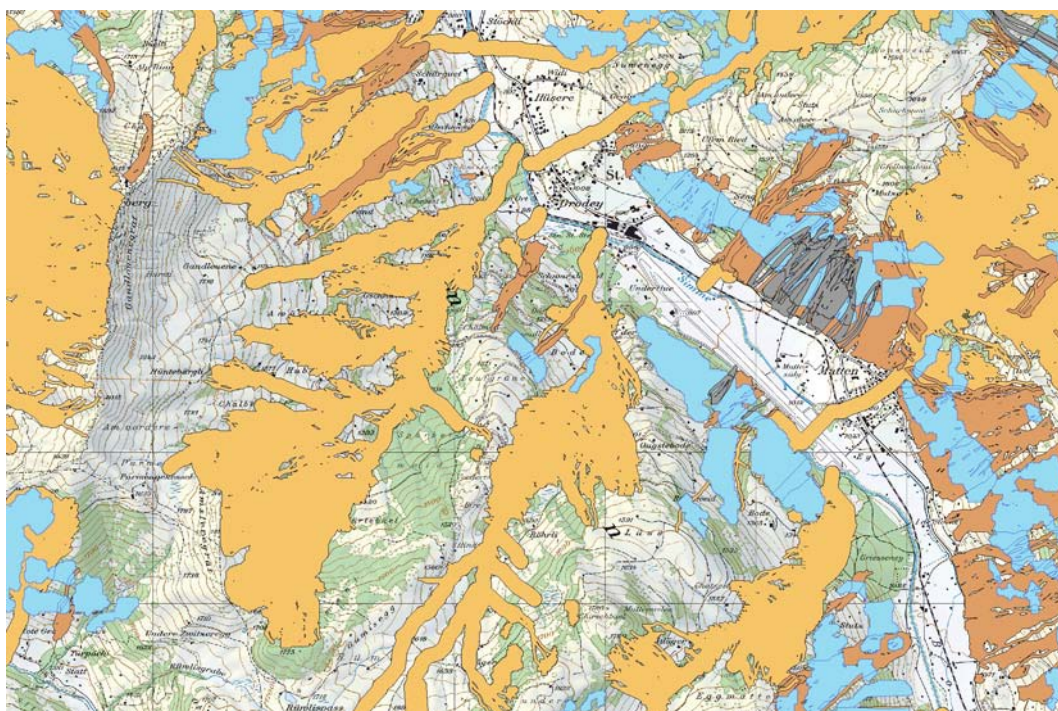


Figure 64: Processus pertinents dans le projet SilvaProtect-CH: avalanche (bleu), chute de pierres (gris), glissement (brun) et processus liés aux cours d'eau (beige).

Dans la tables de ce Shape, il est possible ainsi de trouver les processus détaillés correspondants:

| FID | Shape   | GRS | LAW | STURZ | HM |
|-----|---------|-----|-----|-------|----|
| 122 | Polygon | 0   | 0   | 1     | 0  |
| 123 | Polygon | 0   | 0   | 1     | 0  |
| 124 | Polygon | 0   | 0   | 1     | 0  |
| 125 | Polygon | 0   | 0   | 1     | 0  |
| 126 | Polygon | 0   | 0   | 1     | 0  |
| 127 | Polygon | 0   | 0   | 1     | 0  |
| 128 | Polygon | 0   | 0   | 1     | 0  |
| 129 | Polygon | 0   | 0   | 1     | 0  |
| 130 | Polygon | 0   | 0   | 1     | 0  |
| 131 | Polygon | 0   | 0   | 1     | 0  |
| 132 | Polygon | 0   | 0   | 1     | 0  |
| 133 | Polygon | 0   | 0   | 1     | 0  |
| 134 | Polygon | 0   | 0   | 1     | 0  |
| 135 | Polygon | 0   | 0   | 1     | 0  |
| 136 | Polygon | 0   | 0   | 1     | 0  |
| 137 | Polygon | 0   | 0   | 1     | 0  |
| 138 | Polygon | 0   | 0   | 1     | 0  |
| 139 | Polygon | 0   | 0   | 1     | 0  |
| 140 | Polygon | 0   | 0   | 1     | 0  |
| 141 | Polygon | 0   | 0   | 1     | 0  |
| 142 | Polygon | 0   | 0   | 1     | 0  |
| 143 | Polygon | 0   | 0   | 1     | 0  |
| 144 | Polygon | 0   | 0   | 1     | 0  |
| 145 | Polygon | 0   | 0   | 1     | 0  |
| 146 | Polygon | 0   | 0   | 1     | 0  |

Figure 65: Processus détaillés.

## 6 Surface forestière (SILVA)

Afin de modéliser les processus pertinents en forêt, il est nécessaire d'avoir d'un côté les surfaces des processus pertinents et d'un autre côté la surface forestière. Les premières ont été modélisées dans le chapitre 5 Processus pertinents (INTERSECT), mais les deuxièmes doivent encore être faites.

Les données les plus homogènes pour l'élaboration de la surface forestière proviennent des données du Vecteur25. A cela s'ajoute les surfaces des tempêtes Lothar et Viviane, qui ne sont pas toujours recensées dans le Vecteur25.

Les données utilisées sont:

- Surface primaire Vecteur25:
  - o Forêt
  - o Forêt clairsemée
  - o Pierrier en forêt
  - o Pierrier en forêt clairsemée
  - o Marais en forêt
  - o Marais en forêt clairsemée
- Vivian90: toutes les surfaces
- Lothar99: toutes les surfaces

La catégorie "buisson" n'est pas pris en considération dans les surfaces forestières. En effet, d'une part cette information n'est pas toujours précise et d'autre part, les buissons ne sont pas considérés comme forêt protectrice (voir chapitre 1.2 du rapport principal).

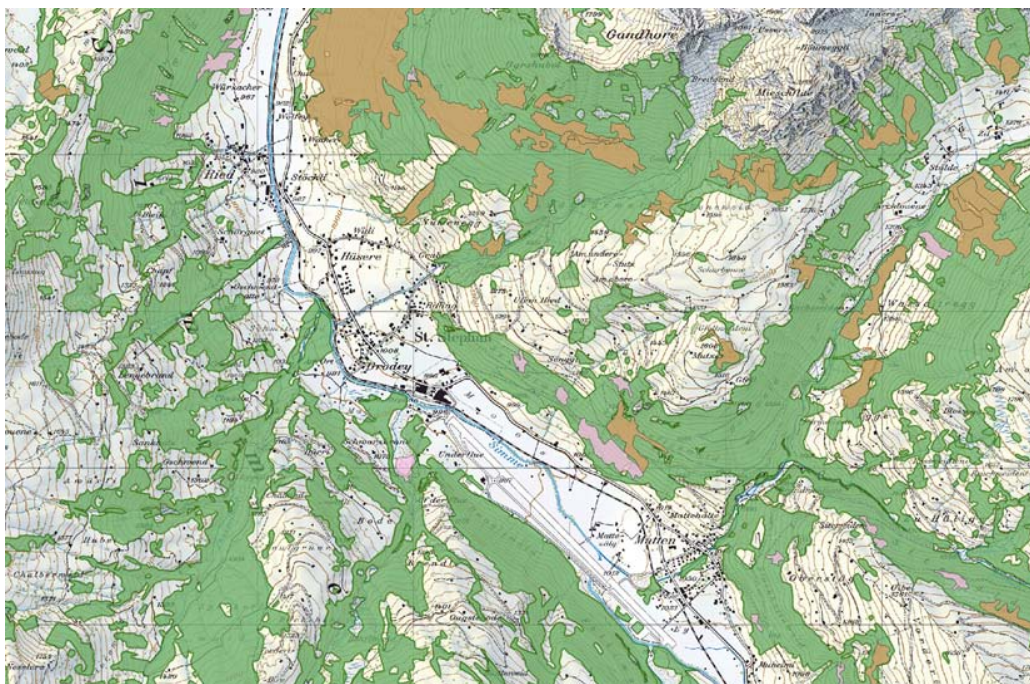
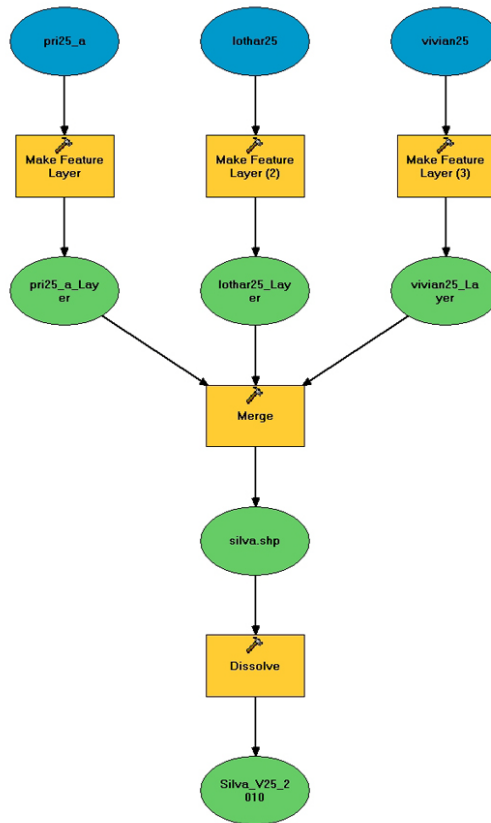


Figure 66: Surfaces forestières du Vecteur25 (vert), surfaces Viviane (violet) et surfaces Lothar (brun).

**Modèle:** Silva (Toolbox: SiPro 93 Silva)

**Input:** pri25\_a, lothar25 et vivian25

**Output:** Silva\_V25.



Modèle 37: Surface forestière.

### Détails du modèle

#### Vector25 surfaces primaires

- **pri25\_a: Make Feature Layer:** "OBJECTVAL" = 'Z\_GerWa' OR "OBJECTVAL" = 'Z\_GerWaO' OR "OBJECTVAL" = 'Z\_SumWa' OR "OBJECTVAL" = 'Z\_SumWaO' OR "OBJECTVAL" = 'Z\_Wald' OR "OBJECTVAL" = 'Z\_WaldOf'

#### Vivian90

- **Vivian25: Make Feature Layer**

#### Lothar99

- **Lothar25: Make Feature Layer**
- **Merge:** Input: pri25\_a\_Layer, lothar25\_Layer et vivian25\_Layer
- **Dissolve:** Create Multipart Feature: non checked



## 7 Indices de forêts protectrices et du potentiel de danger (SYNTHESE)

Le module SYNTHESE comprend les résultats finaux de la modélisation:

- **Indice des forêts protectrices:** recoupe des processus pertinents (INTERSECT) et de la surface forestière (SILVA).
- **Indice du potentiel de dommages:** recoupe les processus pertinents (INTERSECT) et l'enjeux (DAMAGE).

### 7.1 Indices des forêts protectrices

## Description et résultats de la modélisation

L'intersection entre les processus pertinents et la surface forestière donne comme résultat les processus pertinents en termes de dommages en forêts. Cette modélisation se fait par canton; en effet, la grande quantité de données traitées ne permet pas de calculer cela pour toute la Suisse (limite de la machine).

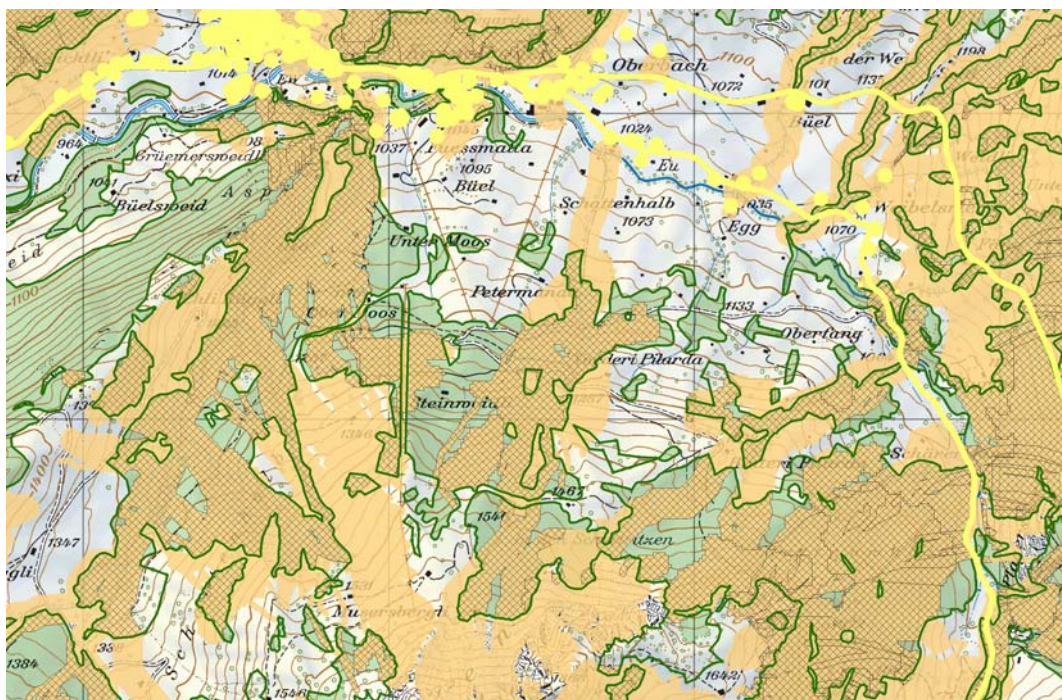
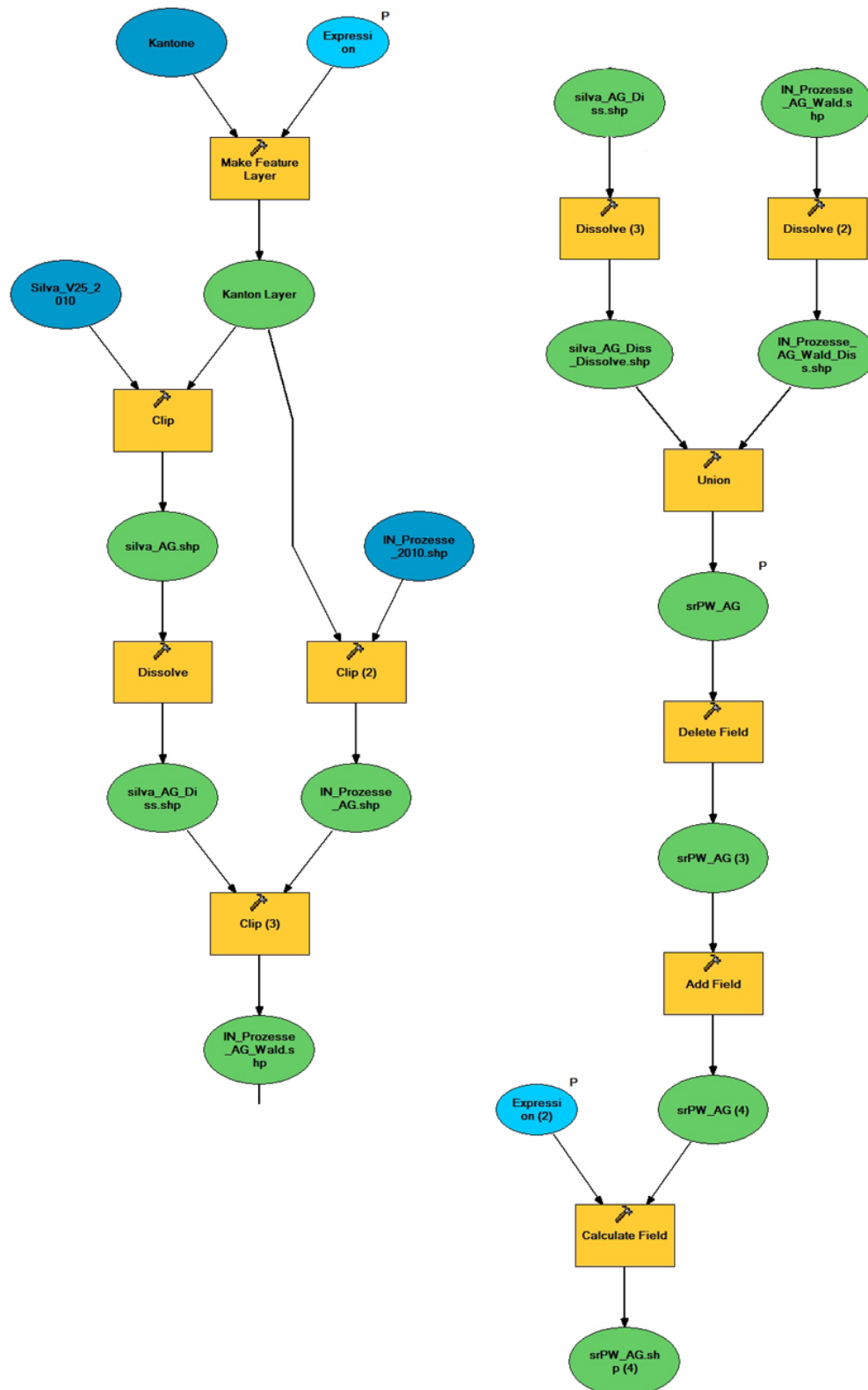


Figure 67: Processus pertinents en forêt (hachurés noirs); enjeux (jaune), processus pertinents (beige) et forêt (vert).

**Modèle:** Schutzwaldindex (Toolbox: SiPro 93 SYNTHESE)

**Input:** Kantone, Silva\_V25 et IN\_Prozesse\_CH.

**Output:** srPW\_AG jusqu'à srPW\_ZH.



Modèle 38: Calcul de l'indice des forêts protectrices.

## Détails du modèle

- **Kantone:** Make Feature Layer: **Expression:** (Model Parameter); "KT" = 'AG' (Figure 68: Expression).

### Surface forestière par canton

- **Clip:** Input: **Silva\_V25\_2010**; Clip: Kanton Layer
- **Dissolve:** Input: Silva\_AG; Create Multipart Feature: non checked
- **Dissolve (3):** Input: Silva\_AG\_Diss; Create Multipart Feature: checked

### Processus pertinents par canton

- **Clip (2):** Input: **IN\_Prozesse\_2010**; Clip: Kanton Layer

### Processus pertinents en forêt

- **Clip (3):** Input: IN\_Prozesse\_AG; Clip: Silva\_AG\_Diss
- **Dissolve (2):** Input: IN\_Prozesse\_AG\_Wald; Dissolve Field(s): GRS, LAW, STURZ et HM: checked; Create Multipart Feature: checked
- **Union:** Input Features: IN\_Prozesse\_AG\_Wald\_diss und Silva\_AG\_Diss\_Dissolve
- **Delete Field:** tous les champs inutiles sont supprimés.
- **Add Field:** Input Table: srPW\_AG; Field Name: ,KT'; Field Type: ,TEXT
- **Calculate Field:** Input Table: srPW\_AG; Field Name: ,KT'; Expression (Figure 68: Expression 2).
- **Expression:** Expression = 'AG'

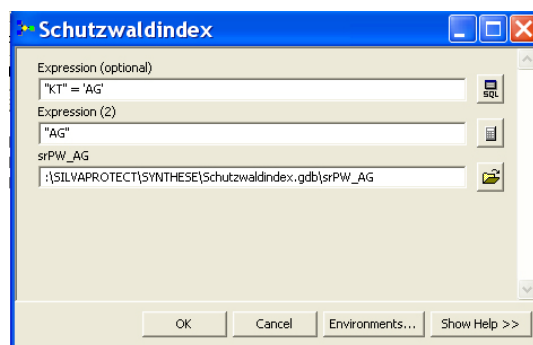


Figure 68: Informations nécessaires pour le calcul de l'IFP.

Le résultat final est livré sous forme de shapefile. Les attributs de la table (Figure 69) informent sur les processus présents dans le canton et sur l'importance de ces processus.

| OBJECTID ^ | Shape ^ | GRS | LAW | STURZ | HM | Shape_Length   | Shape_Area       | KT |
|------------|---------|-----|-----|-------|----|----------------|------------------|----|
| 1          | Polygon | 0   | 0   | 0     | 0  | 4883972.324093 | 448940162.455029 | AG |
| 2          | Polygon | 0   | 0   | 0     | 1  | 717525.401978  | 13811676.929947  | AG |
| 3          | Polygon | 0   | 0   | 1     | 0  | 14274.430863   | 85981.80829      | AG |
| 4          | Polygon | 0   | 0   | 1     | 1  | 26956.686093   | 267694.555379    | AG |
| 5          | Polygon | 1   | 0   | 0     | 0  | 1014824.355163 | 27375658.399555  | AG |
| 6          | Polygon | 1   | 0   | 0     | 1  | 190512.915514  | 3099365.738775   | AG |
| 7          | Polygon | 1   | 0   | 1     | 0  | 3823.555448    | 16504.800511     | AG |
| 8          | Polygon | 1   | 0   | 1     | 1  | 8905.50365     | 104372.923296    | AG |

Figure 69: Attributs du Shape "Processus pertinents en forêt" pour le canton AG.

Cette modélisation est effectuée pour tous les cantons. Tous les Shapes obtenus sont ensuite regroupés (MERGE) en un seul **srPW\_CH** et exportés dans un tableau EXCEL. A l'aide d'un tableau croisé dynamique, l'indice des forêts protectrices est calculé pour chaque canton.



## 7.2 Indice du potentiel de dommages

Pour le calcul de l'indice du potentiel de dommages, tous les enjeux touchés par un processus sont considérés. Les enjeux sont classés selon leur pondération (Figure 70) et les processus considérés sont les avalanches, les laves torrentielles, les chutes de pierres et les débordements avec dépôts.

Cette modélisation se fait par canton; en effet, la grande quantité de données traitées ne permet pas de calculer cela pour toute la Suisse (limite de la machine).

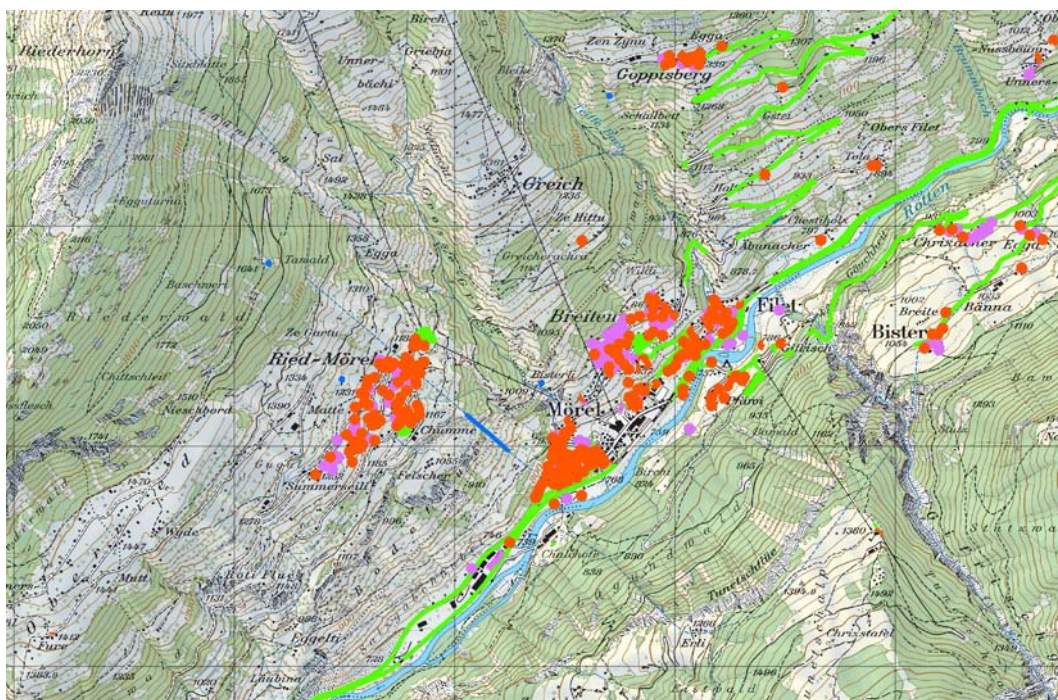
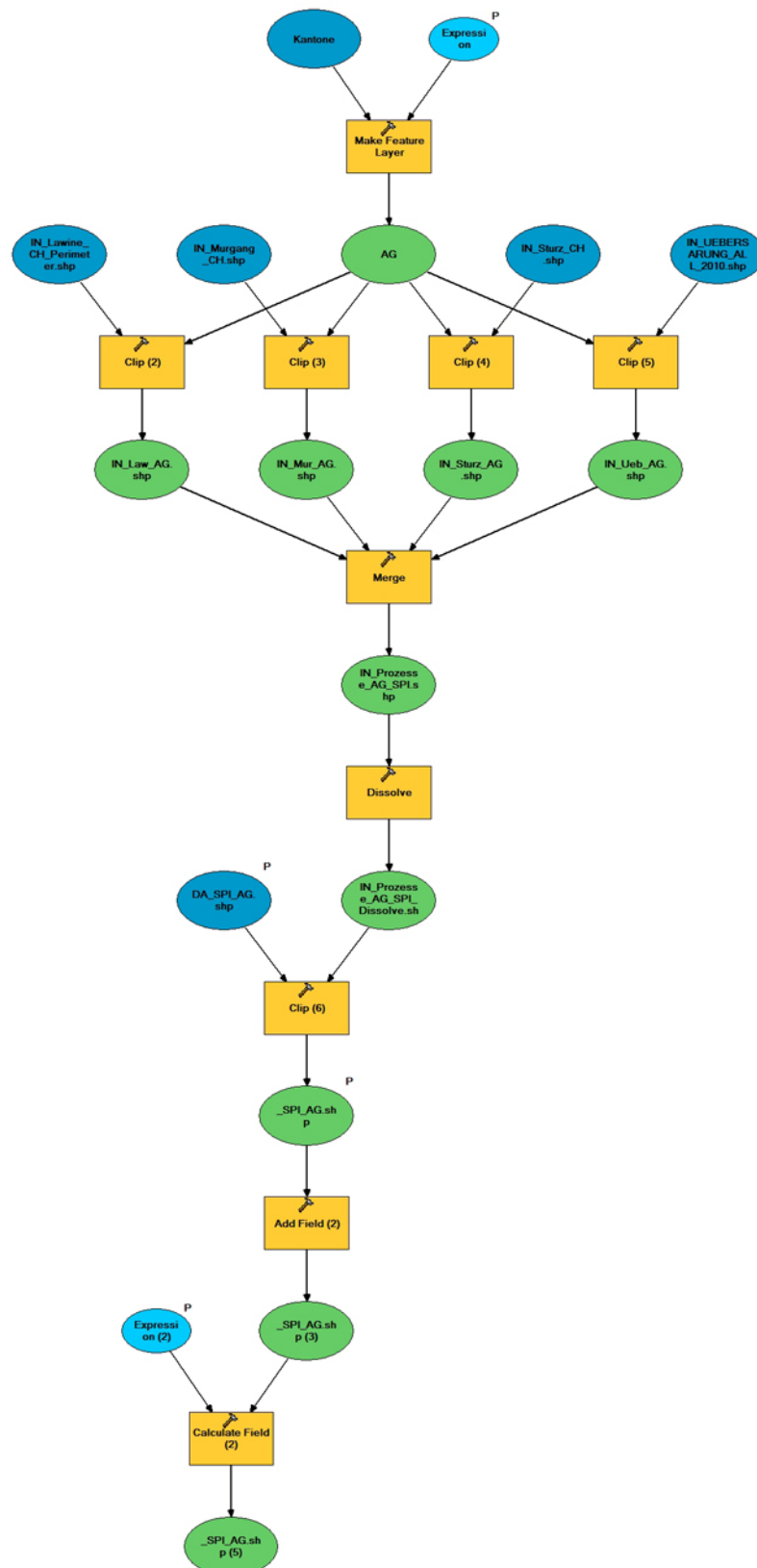


Figure 70: Enjeux touchés par un processus de dangers: pondération 10 (rouge), 5 (violet), 3 (vert) et 1 (bleu).

**Modèle:** Schadenpotentialindex (Toolbox: SiPro 93 SYNTHESE)

**Input:** Kantone, IN\_Lawine\_CH, IN\_Murgang\_CH, IN\_Sturz\_CH et IN\_Uebersarung\_CH..

**Output:** SPI\_AG jusqu'à SPI\_ZH.



Modèle 39: Calcul de l'indice du potentiel de dommages.



## Détails du modèle

- **Kantone:** Make Feature Layer: **Expression:** (Model Parameter); "KT" = 'AG' (Figure 71: Expression).
- **Clip (2):** Input: **IN\_Lawine\_CH**; Clip: Kanton Layer
- **Clip (3):** Input: **IN\_Murgang\_CH**; Clip: Kanton Layer
- **Clip (4):** Input: **IN\_Sturz\_CH**; Clip: Kanton Layer
- **Clip (5):** Input: **IN\_Uebersarung\_CH**; Clip: Kanton Layer
- **Merge:** Input: IN\_Law\_AG, IN\_Mur\_AG, IN\_Sturz\_AG et IN\_Ueb\_AG
- **Dissolve:** Input: IN\_Prozesse\_AG; Create Multipart Feature: non checked
- **Clip (6):** Input: **DA\_SPI\_AG**; Clip: IN\_Prozesse\_AG
- **Add Field:** Input Table: \_SPI\_AG; Field Name: ,KT'; Field Type: ,TEXT
- **Calculate Field:** Input Table: SPI\_AG; Field Name: ,Shape Area'; Expression: Area
- **Calculate Field:** Input Table: SPI\_AG; Field Name: ,KT'; Expression
- **Expression:** Expression = 'AG'

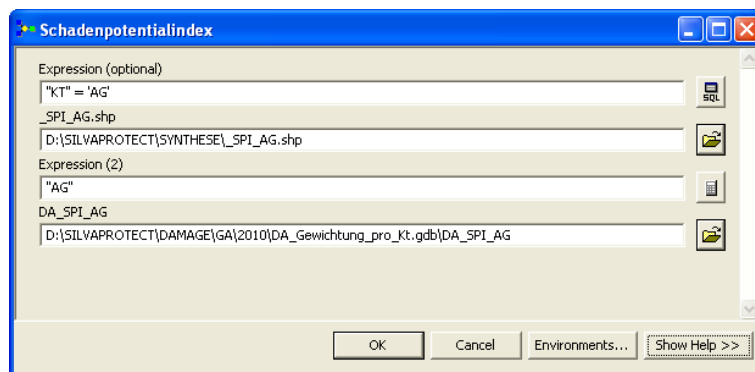


Figure 71: Informations nécessaires pour le calcul de l'IPD.

Grâce à ce modèle, il est possible de calculer la surface de chacun des enjeux touchés par les processus précédemment mentionnés.

| FID | Shape   | GEW | Shape_Leng    | Shape_Area    | KT |
|-----|---------|-----|---------------|---------------|----|
| 0   | Polygon | 1   | 161031.459842 | 3900.099992   | AG |
| 1   | Polygon | 3   | 2998999.93834 | 471487.504701 | AG |
| 2   | Polygon | 5   | 1839576.24083 | 226313.93028  | AG |
| 3   | Polygon | 10  | 4901723.76775 | 2250675.04295 | AG |

Figure 72: Attributs des enjeux pertinents pour un canton.

Cette modélisation est effectuée pour tous les cantons. Tous les Shapes obtenus sont ensuite regroupés (MERGE) en un seul **SPI\_CH** et exportés dans un tableau EXCEL. A l'aide d'un tableau croisé dynamique, l'indice du potentiel de dommages est calculé pour chaque canton.

## 8 Littérature

ESRI, 2008: Une introduction en langue allemande à la construction et au maniement des modèles de géotraitement dans ArcGIS (d), 37 p.

Voir: <http://www.esri-germany.de/downloads/papers/ModelBuilder-Leitfaden.pdf>

ESRI, 2011: ModelBuilder (e), 320 p.

Geo7, 2009: SilvaProtect-CH, Indentification des zones de déclanchement des avalanches (d), 4 p.

Geo7, 2006: SilvaProtect-CH, Modèle de données (d), 16 p.

OFEV, 2008: SilvaProtect-CH Phase I, documentation du projet (d), 242 p.<sup>1</sup>

Swisstopo, 2007: VECTOR25 Le modèle numérique du territoire de la Suisse, 31p.

Voir: <http://www.swisstopo.admin.ch/internet/swisstopo/fr/home/products/landscape/vector25.html>

Swisstopo, 2012: Modèle de données géologiques, 151 p.

Voir: <http://www.geologieportal.ch/internet/geologieportal/fr/home/knowledge/lookup/datamodel.html>

---

<sup>1</sup> Cette publication peut être obtenue à l'adresse email: [gefahrenpraevention@bafu.admin.ch](mailto:gefahrenpraevention@bafu.admin.ch).

## 9 Annexes

### 9.1 Modèle de données

Ce chapitre donne un aperçu du modèle de données pour SilvaProtect-CH et son contenu est décrit de manière détaillée. Il est divisé en deux parties: la première donne un aperçu des données utilisées et des résultats obtenus. Tandis que la deuxième décrit le contenu des données pour tous les modèles utilisés dans les chapitres 4 à 7.

#### 9.1.1 Données utilisées et obtenues

La Figure 73 donne un aperçu des données utilisées et des résultats obtenus dans le projet SilvaProtect-CH. Chaque donnée est décrite de manière détaillée et contient les informations suivantes:

- **Name:** Nom
- **Properties:** Propriété
- **Field Name:** Noms des champs utilisés
- **Value:** Valeur recherchée ou utilisée dans les champs
- **Data Type:** Type de donnée

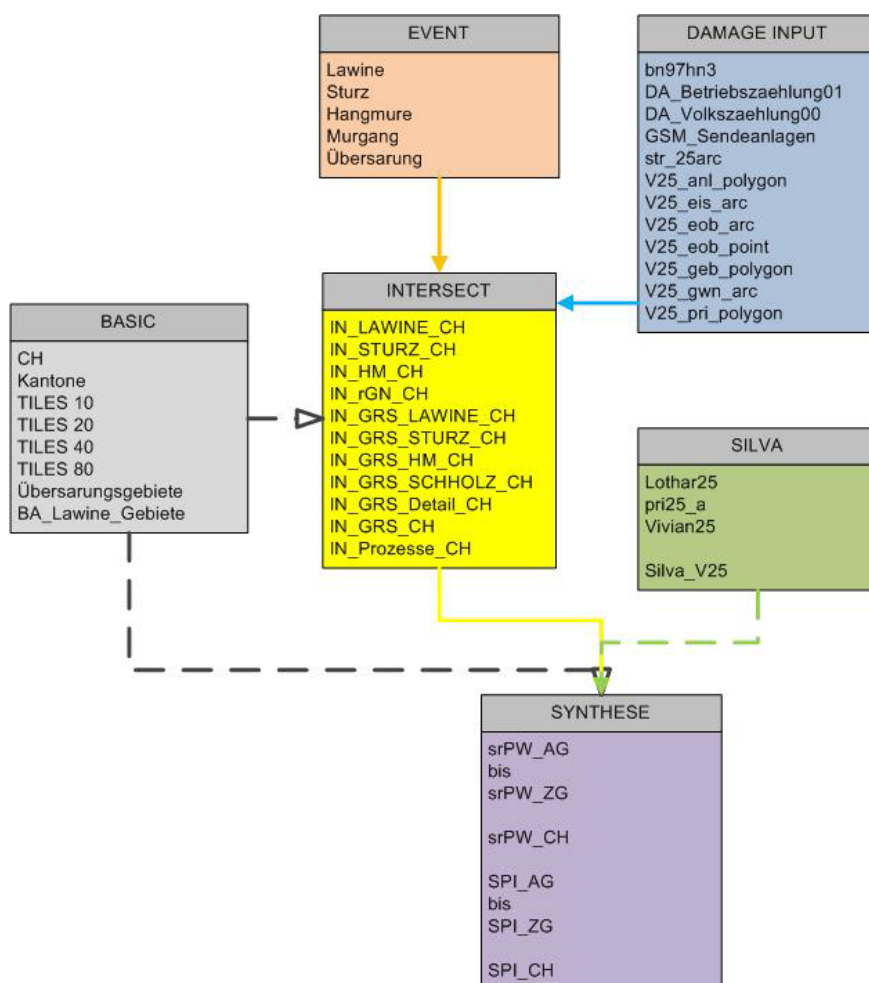


Figure 73: Modèle de données pour SilvaProtect-CH.

## Module BASIC

| Name                | Properties    | Field Name | Value                                                                                                                                                          | Data Type |
|---------------------|---------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| CH                  | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
| Kantone             | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | KT         | AG bis ZG                                                                                                                                                      | Text      |
| TILES 10            | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | TILES      | 1bc, 1bd jusqu'à 49cd                                                                                                                                          | Text      |
| TILES 20            | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | TILES      | 1b, 1d bis 49c                                                                                                                                                 | Text      |
| TILES 40            | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | TILES_NR   | 1 bis 49                                                                                                                                                       | Text      |
|                     |               | DA_GA      | DA_GA_01 jusqu'à DA_GA_49                                                                                                                                      | Text      |
| TILES 80            | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | TILES_NR   | 1 jusqu'à 11                                                                                                                                                   | Text      |
| Uebersarungsgebiete | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | Name       | Aare_Alpin, Inn, Jura,<br>Mittelland_Nord, Mittelland_Sud,<br>Reuss_Linth_Alpin, Rhein_Mitte,<br>Rhein_Alpin, Rhone_Mittelland,<br>Ticino                      | Text      |
| BA_Lawine_Gebiete   | Feature Class | Shape      |                                                                                                                                                                | Polygon   |
|                     |               | Name       | adu, bad, bas, ber, bie, bui, cha,<br>com, dia, eig, fin, kes, lad, leo, lin,<br>mat, muv, obe, pal, pil, rig, rot,<br>sae, ter, tit, toe, tur, umb, wie, wil. | Text      |

## Module EVENT (Avalanche)

| Name                     | Properties    | Feature Class /<br>Item Name | Value | Typology       |
|--------------------------|---------------|------------------------------|-------|----------------|
| xxn_s_ph_r               | Coverage      | arc                          |       | Preliminary    |
| xxn_s_rel_r              |               | Label                        |       | Not Applicable |
| xxn_m_ph_r               |               | polygon                      |       | Exists         |
| xxn_m_rel_r              |               | tic                          |       | Not Applicable |
| xxn_l_ph_r               |               | region.ph                    |       | Exists         |
| xxn_l_rel_r              |               |                              |       |                |
| xxn_l_cap / point        | Coverage      | Shape                        |       | Geometry       |
|                          |               | Area                         |       | Float          |
|                          |               | Perimeter                    |       | Float          |
|                          |               | xxn_l_cap#                   |       | Binary         |
|                          |               | xxn_l_cap-ID                 |       | Binary         |
|                          |               | NameID                       |       | Char           |
| EV_Sturz_01 bis 49       | Feature Class | Shape                        |       | Polyline       |
|                          |               | Shape_Length                 |       | Double         |
| EV_Hangmure_01 bis 49    | Feature Class | Shape                        |       | Polyline       |
|                          |               | Shape_Length                 |       | Double         |
| EV_Murgang_01 bis 49     | Feature Class | Shape                        |       | Polyline       |
|                          |               | Shape_Length                 |       | Double         |
| EV_Uebersarung_01 bis 49 | Feature Class | Shape                        |       | Polyline       |
|                          |               | Shape_Length                 |       | Double         |

## Module DAMAGE (Input)

| Name                  | Properties | Field Name | Value                                                                                                                                                                 | Data Type |
|-----------------------|------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| bn97hn3               | Shapefile  | Shape      |                                                                                                                                                                       | Polygon   |
|                       |            | GRIDCODE   | 15                                                                                                                                                                    | Double    |
| DA_Betriebszaehlung01 | Shapefile  | Shape      |                                                                                                                                                                       | Point     |
|                       |            | METER_X    |                                                                                                                                                                       | Double    |
|                       |            | METER_Y    |                                                                                                                                                                       | Double    |
| DA_Volkszaehlung00    | Shapefile  | X_KOORD    |                                                                                                                                                                       | Double    |
|                       |            | Y_KOORD    |                                                                                                                                                                       | Double    |
|                       |            | A00WDTOT   | 0                                                                                                                                                                     | Double    |
|                       |            | A00WTZTOT  | 0                                                                                                                                                                     | Double    |
| GSM_Sendeanlagen      | Shapefile  | Shape      |                                                                                                                                                                       | Point     |
|                       |            | X          |                                                                                                                                                                       | Double    |
|                       |            | Y          |                                                                                                                                                                       | Double    |
| str_25arc             | Shapefile  | Shape      |                                                                                                                                                                       | Polyline  |
|                       |            | Length     |                                                                                                                                                                       | Double    |
|                       |            | OBJECTVAL  | Autobahn, Autob_Ri, Autostr,<br>Ein_Ausf, A_Zufahrt                                                                                                                   | Text      |
|                       |            | TUNNELTYPE | „ - “                                                                                                                                                                 | Text      |
| V25_anl_polygon       | Shapefile  | Shape      |                                                                                                                                                                       | Polygon   |
|                       |            | Area       |                                                                                                                                                                       | Double    |
| V25_eis_arc           | Shapefile  | Shape      |                                                                                                                                                                       | Polyline  |
|                       |            | Length     |                                                                                                                                                                       | Double    |
|                       |            | OBJECTVAL  | I_Geleis, Gt_Bahn, NS_BAHN1,<br>SS_BAHN1, NS_BAHN2,<br>SS_BAHN2                                                                                                       | Text      |
|                       |            | TUNNELTYPE | „ - “                                                                                                                                                                 | Text      |
| V25_eob_arc           | Shapefile  | Shape      |                                                                                                                                                                       | Polyline  |
|                       |            | Length     |                                                                                                                                                                       | Double    |
|                       |            | OBJECTVAL  |                                                                                                                                                                       | Text      |
| V25_eob_point         | Shapefile  | Shape      |                                                                                                                                                                       | Point     |
|                       |            | OBJECTVAL  | Kamin, Reserv, W_Turm,<br>Antenne, SendeAnl, Hafen,<br>Schiffst, ARA, ElWerk                                                                                          | Text      |
| V25_geb_polygon       | Shapefile  | Shape      |                                                                                                                                                                       | Polygon   |
|                       |            | Area       | 1963.5                                                                                                                                                                | Double    |
|                       |            | OBJECTVAL  | Z_Kuehlturm, Z_Lagertank,<br>Z_WBecken, Z_Gasthof,<br>Z_Gebaeude, Z_Huette<br>Z_Schiessstand, Z_Treibhaus,<br>Z_Innenhof, Z_Kirche, Z_Perron,<br>Z_Schloss, Z_Station | Text      |
| V25_eob_arc           | Shapefile  | Shape      |                                                                                                                                                                       | Polyline  |
|                       |            | Length     |                                                                                                                                                                       | Double    |
|                       |            | OBJECTVAL  | Sender                                                                                                                                                                | Text      |
| V25_gwn_arc           | Shapefile  | Shape      |                                                                                                                                                                       | Polyline  |
|                       |            | OBJECTVAL  | Druckl_1, Druckl_2                                                                                                                                                    | Text      |
| V25_pri_polygon       | Shapefile  | Shape      |                                                                                                                                                                       | Polygon   |
|                       |            | Shape_Area |                                                                                                                                                                       | Double    |
|                       |            | OBJECTVAL  | Z_Siedl                                                                                                                                                               | Text      |



## Module INTERSECT

| Name              | Properties    | Field Name | Value | Data Type     |
|-------------------|---------------|------------|-------|---------------|
| IN_LAWINE_CH      | Shapefile     | Shape      |       | Polygon       |
|                   |               | LAW        | 0, 1  | Short Integer |
| IN_STURZ_CH       | Shapefile     | Shape      |       | Polygon       |
|                   |               | STURZ      | 0, 1  | Short Integer |
| IN_HM_CH          | Shapefile     | Shape      |       | Polygon       |
|                   |               | HM         | 0, 1  | Short Integer |
| IN_rGN_CH         | Feature Class | Shape      |       | Polygon       |
|                   |               | Shape_Area |       | Double        |
| IN_GRS_LAWINE_CH  | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_LAW    | 0, 1  | Double        |
| IN_GRS_STURZ_CH   | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_Sturz  | 0, 1  | Double        |
| IN_GRS_HM_CH      | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_HM     | 0, 1  | Double        |
| IN_GRS_SCHHOLZ_CH | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_SchH   | 0, 1  | Double        |
| IN_GRS_Detail_CH  | Shapefile     | Shape      |       | Polygon       |
|                   |               | GRS_Sturz  | 0, 1  | Short Integer |
|                   |               | GRS_SchH   | 0, 1  | Short Integer |
|                   |               | GRS_LAW    | 0, 1  | Short Integer |
|                   |               | GRS_HM     | 0, 1  | Short Integer |
| IN_GRS_CH         | Shapefile     | Shape      |       | Polygon       |
|                   |               | GRS        | 0, 1  | Short Integer |
| IN_Prozesse_CH    | Shapefile     | Shape      |       | Polygon       |
|                   |               | GRS        | 0, 1  | Short Integer |
|                   |               | LAW        | 0, 1  | Short Integer |
|                   |               | STURZ      | 0, 1  | Short Integer |
|                   |               | HM         | 0, 1  | Short Integer |

## Module SILVA

| Name     | Properties    | Field Name | Value                                                        | Data Type |
|----------|---------------|------------|--------------------------------------------------------------|-----------|
| Lothar25 | Feature Class | Shape      |                                                              | Polygon   |
|          |               | Shape_Area |                                                              | Double    |
| pri25_a  | Feature Class | Shape      |                                                              | Polygone  |
|          |               | Shape_Area |                                                              | Double    |
|          |               | OBJECTVAL  | Z_GerWA, Z_GerWaO,<br>Z_SumWa, Z_SumWaO,<br>Z_Wald, Z_WaldOf | Text      |
| Vivian25 | Feature Class | Shape      |                                                              | Polygon   |
|          |               | Shape_Area |                                                              | Double    |
| pri25_a  | Feature Class | Shape      |                                                              | Polygon   |
|          |               | Shape_Area |                                                              | Double    |

## Module SYNTHESE

| Name    | Properties | Field Name | Value         | Data Type     |
|---------|------------|------------|---------------|---------------|
| srPW_XX | Shapefile  | Shape      |               | Polygon       |
|         |            | GRS        | 0, 1          | Short Integer |
|         |            | LAW        | 0, 1          | Short Integer |
|         |            | STURZ      | 0, 1          | Short Integer |
|         |            | HM         | 0, 1          | Short Integer |
|         |            | Shape_Area |               | Double        |
|         |            | KT         | AG jusqu'à ZH | Text          |
| SPI_XX  | Shapefile  | Shape      |               | Polygon       |
|         |            | Gewichtung | 1, 3, 5, 10   | Short Integer |
|         |            | Area       |               | Double        |
|         |            | KT         | AG jusqu'à ZH | Text          |

## 9.1.2 Description des différents modèles

### 9.1.2.1 Modèles pour l'enjeux

**Modèle: Eisenbahn GA (Toolbox: SiPro 93 DAMAGE) (p. 25)**

Input: V25\_eis\_arc.

Output: DA\_Eisenbahn\_GA\_G1, DA\_Eisenbahn\_GA\_G3, DA\_Eisenbahn\_GA\_G5 et DA\_Eisenbahn\_swi.

| Name               | Properties | Field Name | Value | Data Type     |
|--------------------|------------|------------|-------|---------------|
| DA_Eisenbahn_GA_G1 | Shapefile  | Shape      |       | Polygon       |
|                    |            | Gewichtung | 1     | Short Integer |
|                    |            | DA_Type    | Eis   | Text          |
| DA_Eisenbahn_GA_G3 | Shapefile  | Shape      |       | Polygon       |
|                    |            | Gewichtung | 3     | Short Integer |
|                    |            | DA_Type    | Eis   | Text          |
| DA_Eisenbahn_GA_G5 | Shapefile  | Shape      |       | Polygon       |
|                    |            | Gewichtung | 5     | Short Integer |
|                    |            | DA_Type    | Eis   | Text          |
| DA_Eisenbahn_swi   | Shapefile  | Shape      |       | Polygon       |
|                    |            | DA_Eis     | 1     | Short Integer |

**Modèle: Anlagen GA (Toolbox: SiPro 93 DAMAGE) (p. 27)**

Input: V25\_gwn\_arc, V25\_eob\_point, V25\_eob\_arc, GSM\_Sendeanlagen, V25\_pri\_polygon, V25\_geb\_polygon et V25\_anl\_polygon

Output: DA\_Anlage\_GA\_G1, DA\_Anlage\_GA\_G3, DA\_Anlage\_GA\_G10 et DA\_Anlage\_swi.

| Name             | Properties | Field Name | Value | Data Type     |
|------------------|------------|------------|-------|---------------|
| DA_Anlage_GA_G1  | Shapefile  | Shape      |       | Polygon       |
|                  |            | Gewichtung | 1     | Short Integer |
|                  |            | DA_Type    | Anl   | Text          |
| DA_Anlage_GA_G3  | Shapefile  | Shape      |       | Polygon       |
|                  |            | Gewichtung | 3     | Short Integer |
|                  |            | DA_Type    | Anl   | Text          |
| DA_Anlage_GA_G10 | Shapefile  | Shape      |       | Polygon       |
|                  |            | Gewichtung | 10    | Short Integer |
|                  |            | DA_Type    | Anl   | Text          |
| DA_Anlage_swi    | Shapefile  | Shape      |       | Polygon       |
|                  |            | DA_An      | 1     | Short Integer |

**Modèle: Gebäude GA10 (Toolbox: SiPro 93 DAMAGE) (p. 33)**

Input: DA\_Volkszaehlung00

Output: DA\_Gebauden\_GA\_G10 et DA\_Gebauden\_bew\_swi.

| Name                | Properties | Field Name | Value | Data Type     |
|---------------------|------------|------------|-------|---------------|
| DA_Gebauden_GA_G10  | Shapefile  | Shape      |       | Polygon       |
|                     |            | Gewichtung | 10    | Short Integer |
|                     |            | DA_Type    | Geb   | Text          |
| DA_Gebauden_bew_swi | Shapefile  | Shape      |       | Polygon       |
|                     |            | DA_Geb_bew | 1     | Short Integer |

**Modèle: Gebäude GA5 temp (Toolbox: SiPro 93 DAMAGE) (p. 34)**

Input: DA\_Volkszaehlung00 et V25\_pri\_polygon

Output: DA\_Gebauden\_GA\_G5temp et DA\_Gebauden\_temp\_swi.

| Name                  | Properties | Field Name | Value | Data Type     |
|-----------------------|------------|------------|-------|---------------|
| DA_Gebauden_GA_G5temp | Shapefile  | Shape      |       | Polygon       |
| DA_Gebauden_temp_swi  | Shapefile  | Shape      |       | Polygon       |
|                       |            | DA_Geb_tmp | 1     | Short Integer |

**Modèle: Gebäude GA5 ind punkt (Toolbox: SiPro 93 DAMAGE) (p. 35)**

Input: DA\_Betriebezaehlung et V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G5ind\_pkt.

| Name                     | Properties | Field Name | Value | Data Type |
|--------------------------|------------|------------|-------|-----------|
| DA_Gebauden_GA_G5ind_pkt | Shapefile  | Shape      |       | Polygon   |

**Modèle: Gebäude GA5 ind klein (Toolbox: SiPro 93 DAMAGE) (p. 36)**

Input: DA\_Betriebezaehlung et V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G5ind\_klein.

| Name                       | Properties | Field Name | Value | Data Type |
|----------------------------|------------|------------|-------|-----------|
| DA_Gebauden_GA_G5ind_klein | Shapefile  | Shape      |       | Polygon   |

**Modèle: Gebäude GA5 ind gross (Toolbox: SiPro 93 DAMAGE) (p. 39)**

Input: DA\_Betriebezaehlung et V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G5ind\_gross.

| Name                       | Properties | Field Name | Value | Data Type |
|----------------------------|------------|------------|-------|-----------|
| DA_Gebauden_GA_G5ind_gross | Shapefile  | Shape      |       | Polygon   |

**Modèle: Gebäude GA5 ind areal (Toolbox: SiPro 93 DAMAGE) (p. 41)**

Input: bn97hn3, DA\_Betriebezaehlung, DA\_Volkzaehlung00 et V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G5ind\_areal.

| Name                       | Properties | Field Name | Value | Data Type |
|----------------------------|------------|------------|-------|-----------|
| DA_Gebauden_GA_G5ind_areal | Shapefile  | Shape      |       | Polygon   |

**Modèle: Gebäude GA5 ind (Toolbox: SiPro 93 DAMAGE) (p. 43)**

Input: DA\_gebauden\_GA5 ind pkt, DA\_gebauden\_GA5 ind klein, DA\_gebauden\_GA5 ind\_gross et DA\_gebauden\_GA5 ind areal

Output: DA\_Gebauden\_GA\_G5ind et DA\_Gebauden\_ind\_swi.

| Name                 | Properties | Field Name | Value | Data Type     |
|----------------------|------------|------------|-------|---------------|
| DA_Gebauden_GA_G5ind | Shapefile  | Shape      |       | Polygon       |
| DA_Gebauden_ind_swi  | Shapefile  | Shape      |       | Polygon       |
|                      |            | DA_Ind     | 1     | Short Integer |

**Modèle: Gebäude GA3 oeff (Toolbox: SiPro 93 DAMAGE) (p. 44)**

Input: V25\_geb\_polygon

Output: DA\_Gebauden\_GA\_G3oeff et DA\_Gebauden\_oeff\_swi.

| Name                  | Properties | Field Name | Value | Data Type     |
|-----------------------|------------|------------|-------|---------------|
| DA_Gebauden_GA_G3oeff | Shapefile  | Shape      |       | Polygon       |
| DA_Gebauden_oeff_swi  | Shapefile  | Shape      |       | Polygon       |
|                       |            | DA_Ind     | 1     | Short Integer |



**Modèle: Gebäude GA5 GA3 (Toolbox: SiPro 93 DAMAGE) (p. 45)**

Input: DA\_Gebauden\_GA\_G5ind, DA\_Gebauden\_GA\_G5temp, DA\_Gebauden\_GA\_G3\_oeff et DA\_Gebauden\_GA\_G10

Output: DA\_Gebauden\_GA\_G5 et DA\_Gebauden\_GA\_G3.

| Name              | Properties | Field Name | Value | Data Type     |
|-------------------|------------|------------|-------|---------------|
| DA_Gebauden_GA_G5 | Shapefile  | Shape      |       | Polygon       |
|                   |            | Gewichtung | 5     | Short Integer |
|                   |            | DA_Type    | Geb   | Text          |
| DA_Gebauden_GA_G3 | Shapefile  | Shape      |       | Polygon       |
|                   |            | Gewichtung | 3     | Short Integer |
|                   |            | DA_Type    | Geb   | Text          |

**Modèle: Strassen GA (Toolbox: SiPro 93 DAMAGE) (p. 46)**

Input: Str\_25\_arc

Output: DA\_Strasse\_GA\_G5, DA\_Strasse\_GA\_G3, DA\_Strassen\_swi et DA\_Autobahn\_swi.

| Name             | Properties | Field Name | Value | Data Type     |
|------------------|------------|------------|-------|---------------|
| DA_Strasse_GA_G5 | Shapefile  | Shape      |       | Polygon       |
|                  |            | Gewichtung | 5     | Short Integer |
|                  |            | DA_Type    | Str   | Text          |
| DA_Strasse_GA_G3 | Shapefile  | Shape      |       | Polygon       |
|                  |            | Gewichtung | 3     | Short Integer |
|                  |            | DA_Type    | Str   | Text          |
| DA_Strasse_swi   | Shapefile  | Shape      |       | Polygon       |
|                  |            | DA_Str     | 1     | Short Integer |
| DA_Autobahn_swi  | Shapefile  | Shape      |       | Polygon       |
|                  |            | DA_Autob   | 1     | Short Integer |

**Modèle: DA\_GA (Toolbox: SiPro 93 DAMAGE) (p. 48)**

Input: DA\_Eisenbahn\_swi, DA\_Anlage\_swi, DA\_Gebauden\_bew\_swi, DA\_Gebauden\_temp\_swi, DA\_Gebauden\_ind\_swi, DA\_Gebauden\_oeff\_swi, DA\_Autobahn\_swi et DA\_Strassen\_swi.

Output: DA\_GA\_att et DA\_GA.

| Name      | Properties | Field Name | Value | Data Type     |
|-----------|------------|------------|-------|---------------|
| DA_GA_att | Shapefile  | Shape      |       | Polygon       |
|           |            | DA_Eis     | 0, 1  | Short Integer |
|           |            | DA_Geb_bew | 0, 1  | Short Integer |
|           |            | DA_Geb_tmp | 0, 1  | Short Integer |
|           |            | DA_Ind     | 0, 1  | Short Integer |
|           |            | DA_Geb_oef | 0, 1  | Short Integer |
|           |            | DA_Autob   | 0, 1  | Short Integer |
|           |            | DA_Str     | 0, 1  | Short Integer |
|           |            | DA_AnI     | 0, 1  | Short Integer |
| DA_GA     | Shapefile  | Shape      |       | Polygone      |

**Modèle: DA\_Gew\_Kt\_T1 (Toolbox: SiPro 93 DAMAGE) (p. 50)**

Input: DA\_Eisenbahn\_GA\_G1, DA\_Eisenbahn\_GA\_G3, DA\_Eisenbahn\_GA\_G5, DA\_Anlage\_GA\_G1, DA\_Anlage\_GA\_G3, DA\_Anlage\_GA\_G10, DA\_Gebauden\_GA\_G3, DA\_Gebauden\_GA\_G5, DA\_Gebauden\_GA\_G10, DA\_Strassen\_GA\_G3 et DA\_Strassen\_GA\_G5.

**Modèle: DA\_Gew\_Kt\_T2 (Toolbox: SiPro 93 DAMAGE) (p. 50)**

Input: DA\_GA\_10\_Diss, DA\_GA\_5\_Diss, DA\_GA\_3\_Diss et DA\_GA\_1\_Diss.

**Modèle: DA\_Gew\_Kt (Toolbox: SiPro 93 DAMAGE) (p. 50)**

Input: DA\_Gew\_Kt\_T1 et DA\_Gew\_Kt\_T2.

Output: DA\_SPI\_AG.

| Name      | Properties | Field Name | Value       | Data Type     |
|-----------|------------|------------|-------------|---------------|
| DA_SPI_AG | Shapefile  | Shape      |             | Polygon       |
|           |            | Gewichtung | 1, 3, 5, 10 | Short Integer |

### 9.1.2.2 Modèles pour les processus pertinents

**Modèle: DA\_Lawine\_pro\_Gebiet (Toolbox: SiPro 93 IN Lawine) (p. 57)**

Input: BA\_Lawine\_Gebiete et DA\_GA.

Output: DA\_adu, DA\_bad ... jusqu'à DA\_wil.

| Name   | Properties | Field Name | Value | Data Type |
|--------|------------|------------|-------|-----------|
| DA_adu | Shapefile  | Shape      |       | Polygon   |

**Script: pp\_Setting (p. 59)**

Input: adun\_l\_ph\_r, adun\_l\_rel\_r, adun\_m\_ph\_r, adun\_m\_rel\_r, adun\_s\_ph\_r, adun\_s\_rel\_r, adun\_l\_cap\_point, DHM\_10 et DA\_adu.

Output: adun\_rel\_medium, adun\_rel\_small, adun\_sw\_large. Le résultat final est IN\_LAWINE\_CH.

| Name            | Properties    | Field Name | Value        | Data Type     |
|-----------------|---------------|------------|--------------|---------------|
| adun_rel_medium | Shapefile     | Shape      |              | Polygon       |
|                 |               | NAMEID     | adun_mxx_rel | Text          |
| adun_rel_small  | Shapefile     | Shape      |              | Polygon       |
|                 |               | NAMEID     | adun_sxx_rel | Text          |
| adun_sw_large   | Shapefile     | Shape      |              | Polygon       |
|                 |               | NAMEID     | adun_lxx_rel | Text          |
| IN_LAWINE_CH    | Feature Class | Shape      |              | Polygon       |
|                 |               | LAW        | 1            | Short Integer |

**Modèle: IN Sturz07 (Toolbox: SiPro 93 IN STURZ) (p. 61)**

Input: EV\_Sturz\_01 jusqu'à EV\_Sturz\_49, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49.

Output: 7bd, 7da, 7db et 7dd.

| Name        | Properties | Field Name | Value | Data Type |
|-------------|------------|------------|-------|-----------|
| 5cc bis 5dd | Shapefile  | Shape      |       | Polyline  |

**Modèle: IN STURZ (Toolbox: SiPro 93 IN STURZ) (p. 63)**

Input: IN Sturz01, IN Sturz02, ... et IN Sturz49  
EV\_STURZ, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49.

Output: 1bc, 1bd, jusqu'à 49cc et 49cd. Le résultat final est IN\_STURZ\_CH.

| Name         | Properties    | Field Name | Value | Data Type     |
|--------------|---------------|------------|-------|---------------|
| 1bc bis 49cd | Shapefile     | Shape      |       | Polyline      |
| IN_STURZ_CH  | Feature Class | Shape      |       | Polygon       |
|              |               | STURZ      | 1     | Short Integer |

**Modèle: IN HM01 (Toolbox: SiPro 93 IN HM) (p. 65)**

Input: EV\_HANGMURE\_01 jusqu'à 49, TILES10 et DA\_GA\_01 jusqu'à DA\_GA\_49.

Output: 1bc, 1bd, 1cc, 1cd, 1da, 1db, 1dc et 1dd.

| Name        | Properties | Field Name | Value | Data Type |
|-------------|------------|------------|-------|-----------|
| 1bc bis 1dd | Shapefile  | Shape      |       | Polyline  |

**Modèle: IN HM (Toolbox: SiPro 93 IN HM) (p. 67)**

Input: **IN HM 01, IN HM 02, ... et IN HM 49**  
**EV\_HANGMURE\_01** jusqu'à **49**, **TILES10** et **DA\_GA\_01** jusqu'à **DA\_GA\_49**.

Output: **1bc, 1bd, ... jusqu'à 49cc et 49cd**. Le résultat final est **IN\_HM\_CH**.

| Name                | Properties    | Field Name | Value | Data Type     |
|---------------------|---------------|------------|-------|---------------|
| <b>1bc bis 49cd</b> | Shapefile     | Shape      |       | Polyline      |
| <b>IN_HM_CH</b>     | Feature Class | Shape      |       | Polygon       |
|                     |               | HM         | 1     | Short Integer |

**Modèle: IN\_Ueb (Toolbox: SiPro 93 IN\_Ueb) (p.70)**

Input: **IN HM 01, IN HM 02, ... et IN HM 49**  
**EV\_Uebersarung\_xxx, EV\_Uebersarung\_start\_xxx** et **DA\_GA**.

Output: **IN\_Uebersarung\_start\_xxx**.

| Name                            | Properties | Field Name | Value | Data Type    |
|---------------------------------|------------|------------|-------|--------------|
| <b>IN_Uebersarung_start_xxx</b> | Shapefile  | Shape      |       | Point        |
|                                 |            | ID         |       | Long Integer |
|                                 |            | X          |       | Long Integer |
|                                 |            | Y          |       | Long Integer |

**Modèle: IN Mur 25 (Toolbox: SiPro 93 IN MUR) (p. 76)**

Input: **EV\_MURGANG\_01 bis EV\_MURGANG\_49, TILES10** et **DA\_GA\_01 bis DA\_GA\_45**.

Output: **25aa, 25ab, 25ac, 25ad, 25ba** jusqu'à **25dd**.

| Name                     | Properties | Field Name | Value | Data Type |
|--------------------------|------------|------------|-------|-----------|
| <b>25aa jusqu'à 25dd</b> | Shapefile  | Shape      |       | Polyline  |

**Modèle: IN MUR (Toolbox: SiPro 93 Murgang) (p. 78)**

Input: **IN Mur 01, IN Mur 02, ... et IN Mur 49**  
**EV\_MURGANG\_01 bis EV\_MURGANG\_49, TILES10** et **DA\_GA\_01** jusqu'à **DA\_GA\_49**.

Output: **1bc, 1bd, ... jusqu'à 49cc et 49cd**. Le résultat final est **IN\_MUR\_CH**.

| Name                    | Properties    | Field Name | Value | Data Type     |
|-------------------------|---------------|------------|-------|---------------|
| <b>1bc jusqu'à 49cd</b> | Shapefile     | Shape      |       | Polyline      |
| <b>IN_MUR_CH</b>        | Feature Class | Shape      |       | Polygon       |
|                         |               | HM         | 1     | Short Integer |

### Détermination des lits pertinents (p. 80)

Output: IN\_RUENSE\_UEB\_CH, IN\_RUNSE\_MUR\_CH et IN\_rGN\_CH.

| Name             | Properties | Field Name | Value | Data Type |
|------------------|------------|------------|-------|-----------|
| IN_RUENSE_UEB_CH | Shapefile  | Shape      |       | Polyline  |
| IN_RUNSE_MUR_CH  | Shapefile  | Shape      |       | Polyline  |
| IN_rGN_CH        | Shapefile  | Shape      |       | Polyline  |

### Processus pour les lits pertinentes (p. 84)

Les résultats finaux sont IN\_GRS\_LAWINE\_CH, IN\_GRS\_STURZ\_CH, IN\_GRS\_HM\_CH, IN\_GRS\_SchHolz\_CH, IN\_GRS\_Detail\_CH et IN\_GRS\_CH.

| Name              | Properties    | Field Name | Value | Data Type     |
|-------------------|---------------|------------|-------|---------------|
| IN_GRS_Lawine_CH  | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_Law    | 1     | Short Integer |
| IN_GRS_Sturz_CH   | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_Sturz  | 1     | Short Integer |
| IN_GRS_HM_CH      | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_HM     | 1     | Short Integer |
| IN_GRS_SchHolz_CH | Feature Class | Shape      |       | Polygon       |
|                   |               | GRS_SchH   | 1     | Short Integer |
| IN_GRS_Detail_CH  | Shapefile     | Shape      |       | Polygon       |
|                   |               | GRS_Sturz  | 0, 1  | Short Integer |
|                   |               | GRS_SchH   | 0, 1  | Short Integer |
|                   |               | GRS_Law    | 0, 1  | Short Integer |
|                   |               | GRS_HM     | 0, 1  | Short Integer |
| IN_GRS_CH         | Shapefile     | Shape      |       | Polygon       |
|                   |               | GRS        | 1     | Short Integer |

### Groupeement des processus pertinents (p. 86)

Input: IN\_Lawine\_CH, IN\_Sturz\_CH, IN\_HM\_CH et IN\_GRS\_CH

Output: IN\_Prozesse\_CH.

| Name           | Properties | Field Name | Value | Data Type     |
|----------------|------------|------------|-------|---------------|
| IN_Prozesse_CH | Shapefile  | Shape      |       | Polygon       |
|                |            | GRS        | 0, 1  | Short Integer |
|                |            | LAW        | 0, 1  | Short Integer |
|                |            | STURZ      | 0, 1  | Short Integer |
|                |            | HM         | 0, 1  | Short Integer |



### 9.1.2.3 Modèle pour la surface forestière

**Modèle:** **Silva** (Toolbox: **SiPro 93 Silva**) (p. 87)

Input: [pri25\\_a](#), [lothar25](#) et [vivian25](#).

Output: [Silva\\_V25](#).

| Name      | Properties    | Field Name | Value | Data Type |
|-----------|---------------|------------|-------|-----------|
| Silva_V25 | Feature Class | Shape      |       | Polygon   |

### 9.1.2.4 Modèles pour les indices

**Modèle:** **Schutzwaldindex** (Toolbox: **SiPro 93 SYNTHES**) (p. 89)

Input: [Kantone](#), [Silva\\_V25\\_2010](#) et [IN\\_Prozesse\\_2010](#).

Output: [srPW\\_AG](#) jusqu'à [srPW\\_ZH](#) et [srPW\\_CH](#).

| Name                       | Properties | Field Name | Value         | Data Type     |
|----------------------------|------------|------------|---------------|---------------|
| srPW_AG jusqu'à<br>srPW_ZH | Shapefile  | Shape      |               | Polygon       |
|                            |            | GRS        | 0, 1          | Short Integer |
|                            |            | LAW        | 0, 1          | Short Integer |
|                            |            | STURZ      | 0, 1          | Short Integer |
|                            |            | HM         | 0, 1          | Short Integer |
|                            |            | KT         | AG jusqu'à ZH | Text          |
| srPW_CH                    | Shapefile  | Shape      |               | Polygon       |
|                            |            | GRS        | 0, 1          | Short Integer |
|                            |            | LAW        | 0, 1          | Short Integer |
|                            |            | STURZ      | 0, 1          | Short Integer |
|                            |            | HM         | 0, 1          | Short Integer |
|                            |            | KT         | AG jusqu'à ZH | Text          |

**Modèle:** **Schadenpotentialindex** (Toolbox: **SiPro 93 SYNTHES**) (p. 92)

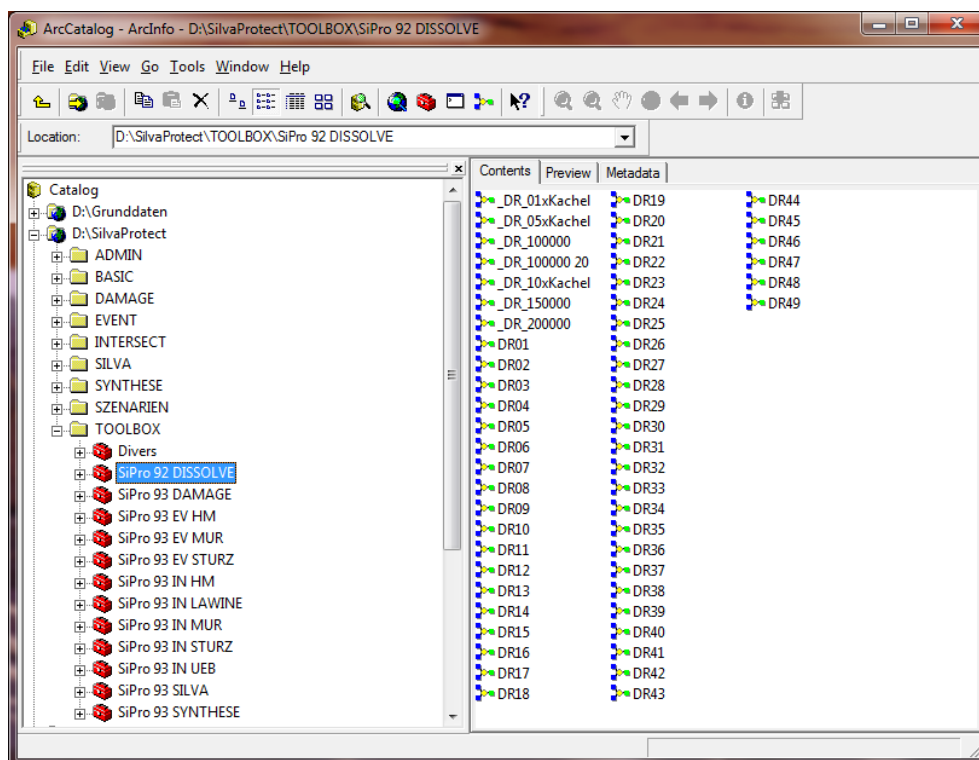
Input: [Kantone](#), [IN\\_Lawine\\_CH](#), [IN\\_Murgang\\_CH](#), [IN\\_Sturz\\_CH](#) et [IN\\_Uebersarung\\_CH](#).

Output: [SPI\\_AG](#) et [SPI\\_CH](#).

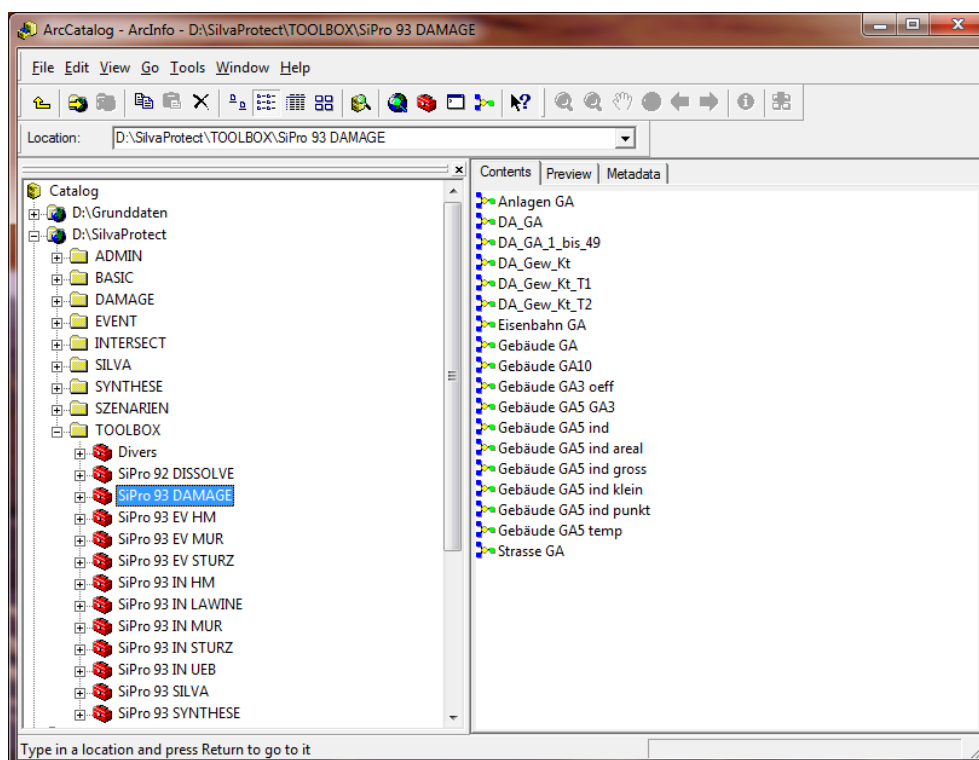
| Name                  | Properties | Field Name | Value         | Data Type     |
|-----------------------|------------|------------|---------------|---------------|
| SPI_AG jusqu'à SPI_ZH | Shapefile  | Shape      |               | Polygon       |
|                       |            | Gewichtung | 1, 3, 5, 10   | Short Integer |
|                       |            | KT         | AG jusqu'à ZH | Text          |
| SPI_CH                | Shapefile  | Shape      |               | Polygon       |
|                       |            | Gewichtung | 1, 3, 5, 10   | Short Integer |
|                       |            | KT         | AG jusqu'à ZH | Text          |

## 9.2 Aperçu des modèles de traitement de ModelBuilder

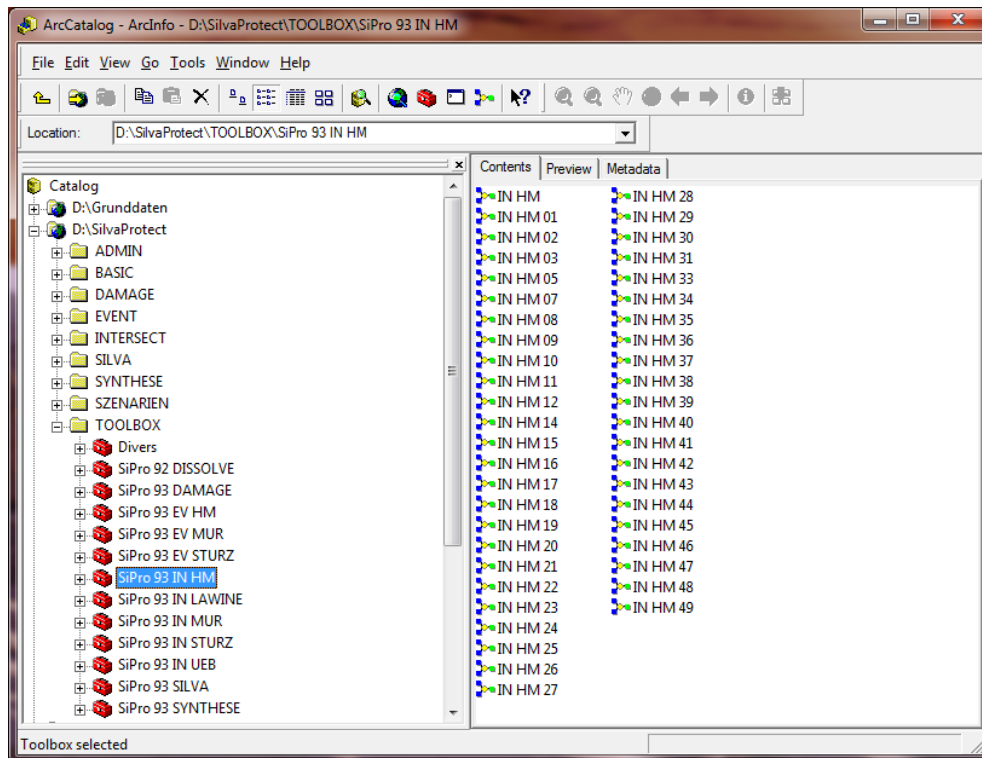
### SiPro 92 Dissolve



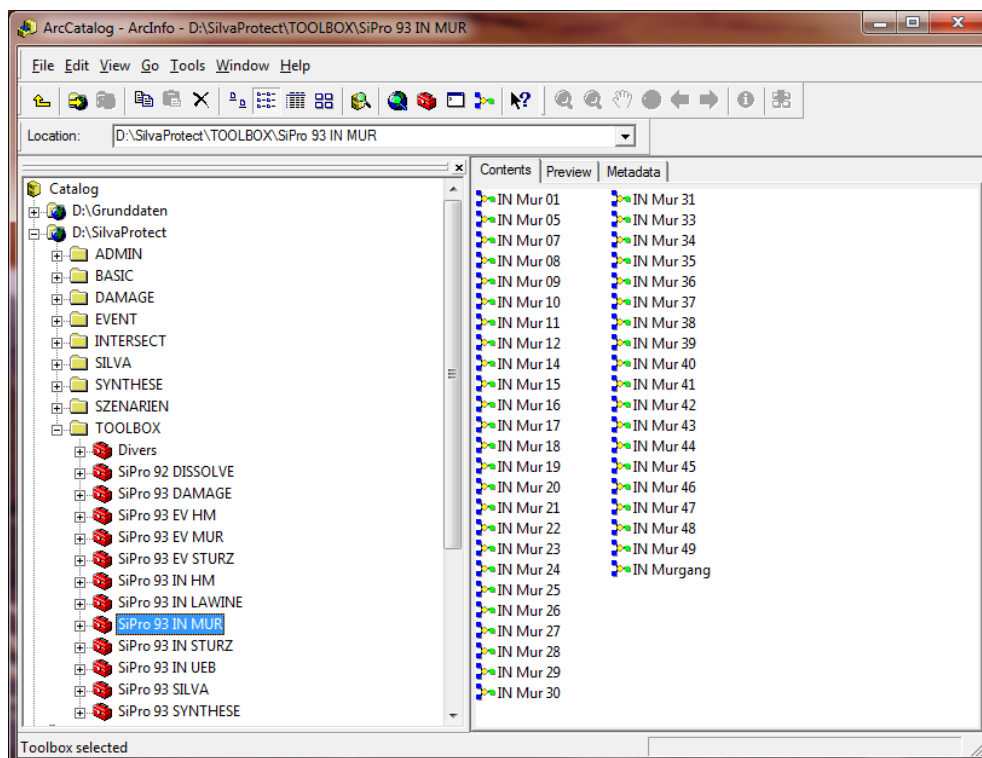
### SiPro 93 DAMAGE



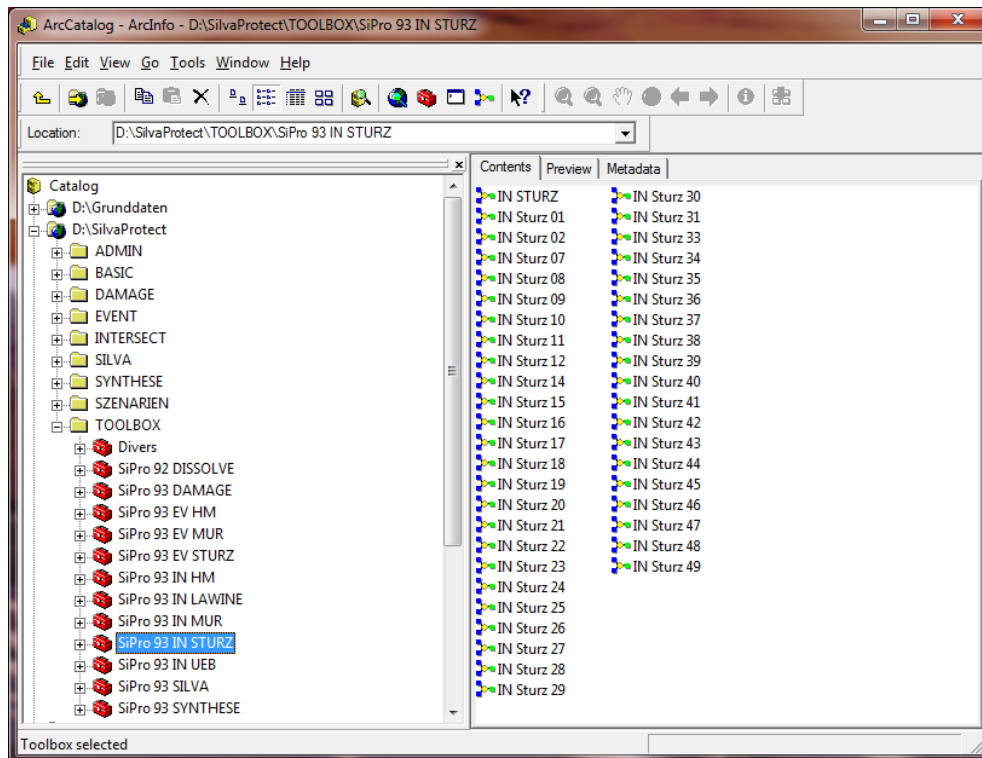
## SiPro 93 IN HM



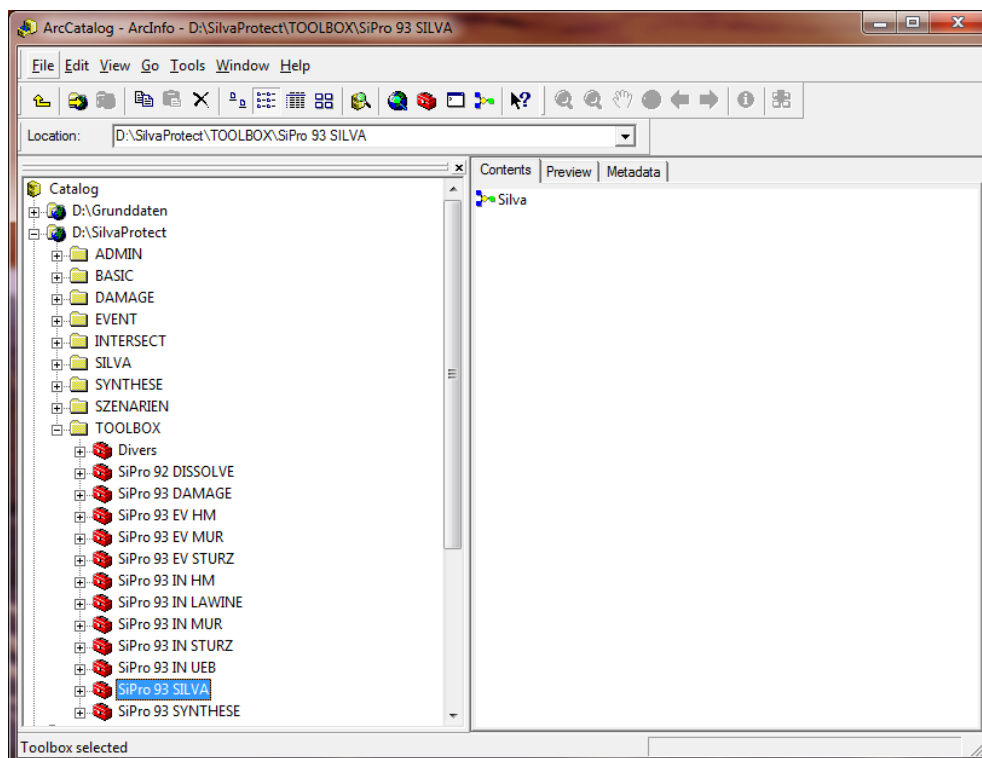
## SiPro 93 IN MUR



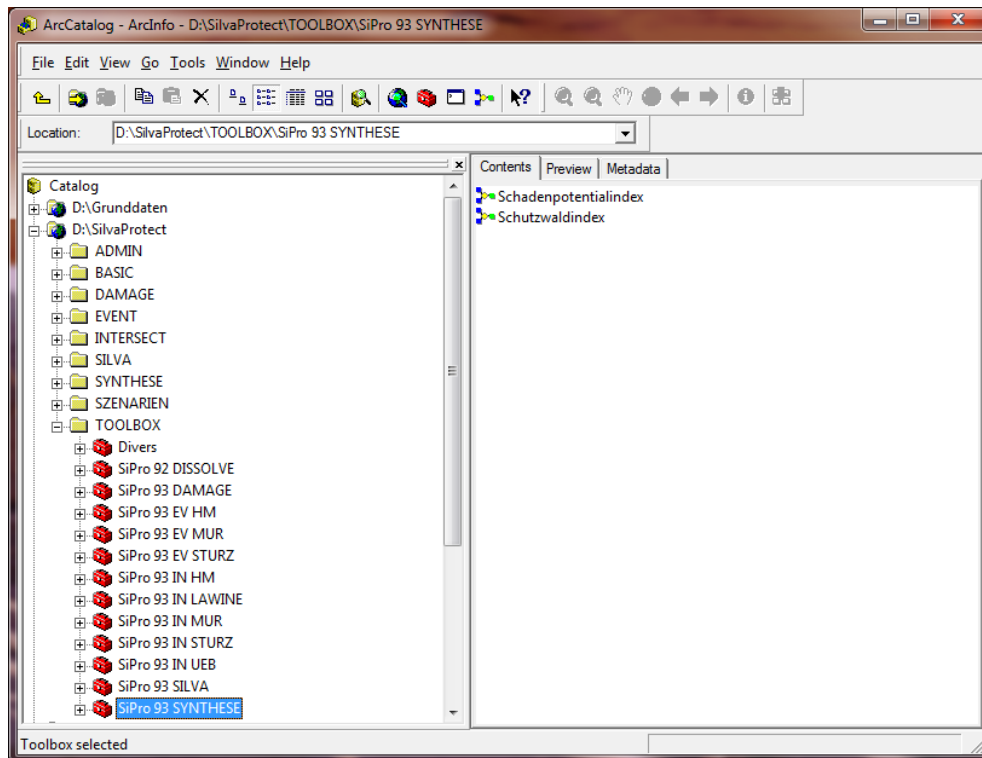
## SiPro 93 IN STURZ



## SiPro 93 SILVA



## SiPro 93 SYNTHESE





## 9.3 Script Python pour les avalanches

### 9.3.1 Pp\_Launch.py: départ

```
#!/usr/bin/env python
# coding: utf-8 -*-
# pp_Launch.py
"""
@summary: Identifikation schadenpotenzialsrelevanter Lawinenanrissgebiete Silvaproduct.

Description:
Identifikation der schadenpotenzialsrelevanten Lawinenperimeter(Silvaproduct PH)
und der dazugehörigen Anrissgebiete (Silvaproduct REL)
für die Kategorien small, medium und large.

History:
Postprocessing_small_med_large_v2_04.py WSL
Migration nach ArcGIS 9.1 geo7
Migration nach ArcGIS 9.3.1 geo7

Family:
pp_Launch.py (aufrufendes Skript)
pp_Program.py (Ausführung einzelner Verarbeitungsschritt)
pp_proc_rel.py (alle Verarbeitungsschritte und Methoden)
pp_Settings.py (Parametrisierung und Einstellungen)
pp_util_GIS.py (Aufruf gp-Objekt, Benachrichtigungen)

Dokumentation:
Workflow und Produkte sind beschrieben in:
Identifikation schadenpotenzialsrelevanter Lawinenanrissgebiete Silvaproduct
geo7, 04.7.2011

@version: 20110704 2.0
@author: Peter Gsteiger (gsp), Andri Baltensweiler (WSL) Originalscript
@note: Der Ablauf ist in __main__ abgebildet.
@note: Wird das Skript ohne Parameter aufgerufen, werden die Parameter in pp_Settings.py gelesen.

@param1: src_Workspace (workspace), Ablage der Coverages mit PH und REL regions
@param2: tar_Folder (Folder), Zielverzeichnis (lokal) zur Ablage der Ergebnisse
@param3: gebiet, (String,3 Buchstaben), Gebietskürzel
@param4: DamLayer (Feature Class), Schadenpotenzial, Polygone)
@param5: dtm (Rasterdataset), DTM Silvaproduct

"""

# ----- Import system modules
import sys
import os
import codecs
import gc
import time,datetime

# ----- Import delivered modules
import pp_Settings
import pp_util_GIS

# ----- Vars
_path_python = pp_Settings._path_python
_path_script = pp_Settings._path_script

# -----
def check_exit(succ_value):
    if succ_value == 1:
        scriptmsg = ">>> Prozess Abbruch " + str(time_tar_gdb) + " in " + Verarbeitung
        pp_util_GIS.gpmsg(gp,scriptmsg,0)
        sys.exit(succ_value)
    else:
        scriptmsg = Verarbeitung + " ok "
        pp_util_GIS.gpmsg(gp,scriptmsg,0)

# ----- Main-Procedure
if __name__ == '__main__':
```

```

'''
'''

# ----- Zeitstempel Verarbeitungsbeginn
starttime = time.clock()
time_tar_gdb = time.strftime("%Y%m%d%H%M%S")
#time_tar_gdb = str(20110704003209)

# ----- Defaults
gp = None
Verarbeitung = ""

# ----- Eingabe Parameter
src_Workspace = "#"
tar_Folder = "#"
gebiet = "#"
DamLayer = "#"
dtm = "#"

try:
    src_Workspace = sys.argv[1]
    tar_Folder = sys.argv[2]
    gebiet = sys.argv[3].lower()
    DamLayer = sys.argv[4]
    dtm = sys.argv[5]
except:
    pass

if src_Workspace == "#":
    src_Workspace = pp_Settings.src_Workspace
if tar_Folder == "#":
    tar_Folder = pp_Settings.tar_Folder
if gebiet == "#":
    gebiet = pp_Settings.gebiet.lower()
if DamLayer == "#":
    DamLayer = pp_Settings.DamLayer
if dtm == "#":
    dtm = pp_Settings.dtm

# ----- Abarbeiten Prozesse
param = [_path_python + "/python.exe", _path_script + "/pp_Program.py", src_Workspace, tar_Folder, gebiet,
DamLayer, dtm, str(time_tar_gdb)]
succ_value = 0
nameidfile = tar_Folder + "/" + gebiet + str(time_tar_gdb) + "/" + gebiet + ".txt"

scriptmsg = ">>> Prozess Start " + str(time_tar_gdb)
pp_util_GIS.gpmsg(gp,scriptmsg,0)

scriptmsg = "Parameter zu Verarbeitung " + str(time_tar_gdb) + ":"
pp_util_GIS.gpmsg(gp,scriptmsg,0)
scriptmsg = "src_Workspace: " + src_Workspace
pp_util_GIS.gpmsg(gp,scriptmsg,0)
scriptmsg = "tar_Folder: " + tar_Folder
pp_util_GIS.gpmsg(gp,scriptmsg,0)
scriptmsg = "gebiet: " + gebiet
pp_util_GIS.gpmsg(gp,scriptmsg,0)
scriptmsg = "DamLayer: " + DamLayer
pp_util_GIS.gpmsg(gp,scriptmsg,0)
scriptmsg = "dtm: " + dtm
pp_util_GIS.gpmsg(gp,scriptmsg,0)

# -----
Verarbeitung = "CREATE_WORKSPACES"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, ""])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "BASISDATEN"
scriptmsg = ">> " + Verarbeitung

```

```

pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, ""])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "PROC_SPOTREL_PH"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, ""])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "RDS_BASIS_GEBIET"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, ""])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "GET_NAMEIDS"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "PROC_NAMEIDS"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    file = open(nameidfile, "r")
    lines = file.readlines()
    file.close()

    anznames = len(lines)
    ncount = 0

    for line in lines:
        ncount = ncount + 1
        nameid = line.replace("\n", "")
        scriptmsg = "verarbeite NAMEID: " + nameid + " " + str(ncount) + " von " + str(anznames)
        pp_util_GIS.gpmsg(gp,scriptmsg,0)

        os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, nameid])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

# -----
Verarbeitung = "PROC_SPOTREL_REL"
scriptmsg = ">> " + Verarbeitung
pp_util_GIS.gpmsg(gp,scriptmsg,0)
try:
    os.spawnv(os.P_WAIT, _path_python + "/python.exe", param + [Verarbeitung, ""])
except Exception, e:
    succ_value = 1
check_exit(succ_value)

if succ_value == 0:
    scriptmsg = ">>> Prozess Ende " + str(time_tar_gdb) + " ok"
    pp_util_GIS.gpmsg(gp,scriptmsg,0)

sys.exit(succ_value)

```

### 9.3.2 Pp\_Programm.py: déclencheur des différentes étapes

```
#!/usr/bin/env python
# coding: utf-8 -*-
#-----
# pp_Program.py

'''
'''

# ----- Import system modules
import sys
import os
import time,datetime

# ----- Import delivered modules
import pp_Settings
import pp_util_GIS
import pp_proc

if __name__ == '__main__':

    # ----- Parameter

    try:
        src_Workspace = str(sys.argv[1])
    except:
        src_Workspace = pp_Settings.src_Workspace

    try:
        tar_Folder = sys.argv[2]
    except:
        tar_Folder = pp_Settings.tar_Folder

    try:
        gebiet = str(sys.argv[3])
    except:
        gebiet = pp_Settings.gebiet.lower()

    try:
        DamLayer = sys.argv[4]
    except:
        DamLayer = pp_Settings.DamLayer

    try:
        dtm = sys.argv[5]
    except:
        dtm = pp_Settings.dtm

    try:
        time_tar_gdb = sys.argv[6]
    except:
        time_tar_gdb = time.strftime("%Y%m%d%H%M%S")

    try:
        Verarbeitung = str(sys.argv[7])
    except:
        Verarbeitung = "CREATE_WORKSPACES"
        Verarbeitung = "BASISDATEN"
        Verarbeitung = "PROC_SPOTREL_PH"
        Verarbeitung = "RDS_BASIS_GEBIET"
        Verarbeitung = "GET_NAMEIDS"
        Verarbeitung = "PROC_NAMEIDS"
        Verarbeitung = "PROC_SPOTREL_REL"
        Verarbeitung = "RDS_BASIS_GEBIET"
        Verarbeitung = "PROC_SPOTREL_PH"

    try:
        nameid = sys.argv[8]
    except:
        nameid = "umbn_l10_ph"

    pp = pp_proc.pp_proc(src_Workspace, tar_Folder, gebiet, DamLayer, dtm, time_tar_gdb, Verarbeitung, nameid)
    pp.Prozesssteuerung()
```

### 9.3.3 Pp\_proc\_rel.py: toutes les étapes de calcul et méthode

```
#!/usr/bin/env python
# coding: utf-8 -*-
'''
pp_proc.py

'''

# ----- Import system modules
import sys, os, time, string, copy

# ----- Import delivered modules
import pp_Settings
import pp_util_GIS

class pp_proc:

    # -----
    def __init__(self, __src_Workspace, __tar_Folder, __gebiet, __DamLayer, __dtm, __time_tar_gdb,
__Verarbeitung, __nameid):

        # ----- Globals
        self.src_Workspace = __src_Workspace
        self.tar_Folder = __tar_Folder
        self.gebiet = __gebiet
        self.DamLayer = __DamLayer
        self.dtm = __dtm
        self.time_tar_gdb = __time_tar_gdb
        self.Verarbeitung = __Verarbeitung
        self.nameid = __nameid

        self.gebiet = self.gebiet.lower()
        self.gebiet = self.gebiet.strip(" ")

        # ----- Vars
        self.ph_string = pp_Settings.ph_string.lower()
        self.rel_string = pp_Settings.rel_string.lower()
        self.cap_string = pp_Settings.cap_string.lower()
        self.wald_string = pp_Settings.wald_string.lower()

        self.tar_gdbname = self.gebiet + self.time_tar_gdb + ".gdb"
        self.tar_gdb = self.tar_Folder + "/" + self.tar_gdbname
        self.rds_folder = self.tar_Folder + "/" + self.gebiet + self.time_tar_gdb

        #self.tar_gdb = r"C:\Workspace\KPRO_LASCRIP2011\umb20110704160856.gdb"
        #self.rds_folder = r"C:\Workspace\KPRO_LASCRIP2011\umb20110704160856"

        self.srcFC_spot = self.tar_gdb + "/" + self.gebiet + "_spot"
        self.srcFC_cap = self.tar_gdb + "/" + self.gebiet + "_cap"
        self.srcFC_ph = self.tar_gdb + "/" + self.gebiet + self.wald_string + "_" + self.ph_string + "_" + "large"
        self.srcFC_rel = self.tar_gdb + "/" + self.gebiet + self.wald_string + "_" + self.rel_string + "_" + "large"

        self.nameidfile = self.rds_folder + "/" + self.gebiet + ".txt"

        self.Aspect = self.rds_folder + "/" + "aspect"
        self.aspRad = self.rds_folder + "/" + "asprad"
        self.aspCosRad = self.rds_folder + "/" + "aspcosrad"
        self.aspSinRad = self.rds_folder + "/" + "aspsinrad"
        self.Slope = self.rds_folder + "/" + "slope"
        self.mainSlpDir = self.rds_folder + "/" + "mainslpdir"
        self.SPgd = self.rds_folder + "/" + "spgd"

        # ----- self.gp-Aufruf und self.gp-Settings
        ARCLICREQ = pp_Settings.ARCLICREQ
        self.gp = None
        ARCGIS_DIR = os.path.expandvars("$ARCGISHOME")
        self.gp = pp_util_GIS.GIS_Setup(ARCLICREQ)

        # Veruegbarkeit self.gp-Objekt
        if self.gp == None:
```



```

        scriptmsg = "Abbruch! Das Geoprocessor Objekt konnte nicht erstellt werden! "
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()
# Pruefen der eingestellten Lizenzierung
else:
    lic_dt = self.gp.ProductInfo()

    if string.lower(ARCLICREQ) <> string.lower(lic_dt):
        scriptmsg = "ArcGIS-Lizenz nicht verfuegbar: " + str(ARCLICREQ)
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()
    else:
        scriptmsg = "ArcGIS-Lizenz: " + str(ARCLICREQ)
        pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

self.gp.OverWriteOutput = 1
pp_util_GIS.gdbsettings(self.gp)
return

# -----
def Prozesssteuerung(self):
    """
    @summary: Steuerung self.Verarbeitung
    @summary: unerwartete Abbrueche werden vom aufrufenden Skript NGKLU_Produkte_PyLaunch.py ab-
    gefangen
    """

    # ----- Validierung Parameter
    # src_Workspace pruefen
    if not self.gp.exists(self.src_Workspace):
        scriptmsg = "Abbruch! Workspace " + self.src_Workspace + " existiert nicht."
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # self.tar_Folder pruefen
    if not os.path.isdir(self.tar_Folder):
        scriptmsg = "Abbruch! Folder " + self.tar_Folder + " existiert nicht."
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # self.DamLayer pruefen
    if not self.gp.exists(self.DamLayer):
        scriptmsg = "Abbruch! Schadenpotenzial " + self.DamLayer + " existiert nicht."
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # self.dtm pruefen
    if not self.gp.exists(self.dtm):
        scriptmsg = "Abbruch! dtm " + self.dtm + " existiert nicht."
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # -----
    if self.Verarbeitung == "CREATE_WORKSPACES":
        self.CREATE_WORKSPACES()

    # -----
    if self.Verarbeitung == "BASISDATEN":
        self.BASISDATEN()

    # -----
    if self.Verarbeitung == "PROC_SPOTREL_PH":
        self.PROC_SPOTREL_PH()

    # -----
    if self.Verarbeitung == "RDS_BASIS_GEBIET":
        self.RDS_BASIS_GEBIET()

    # -----
    if self.Verarbeitung == "GET_NAMEIDS":
        self.GET_NAMEIDS()

```

```

# -----
if self.Verarbeitung == "PROC_NAMEIDS":
    self.PROC_NAMEIDS()

# -----
if self.Verarbeitung == "PROC_SPOTREL_REL":
    self.PROC_SPOTREL_REL()

del self.gp
return

# -----
def PROC_SPOTREL_REL(self):
    scriptmsg = "berechne schadenpotenzialsrelevante Anrissgebiete REL "
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

    #find phSuffix in the first fc
    self.gp.workspace = self.tar_gdb
    searchstring = "*_ws"
    fcs = self.gp.ListFeatureClasses(searchstring,"POLYGON")
    wsfclist = []
    for entry in fcs:
        wsfclist.append(entry)

    # Die berechneten Einzugsgebiete werden mit den zugehoerigen rel Features verschnitten.
    # Liste der Einzugsgebiete erstellen

    count = 0
    swFClist = []
    appendexpr = ""

    fielddef_nameidws = ["nameidws","TEXT","","","","","NULLABLE","NON_REQUIRED",""]
    fielddef_relOK = ["relOK","SHORT","","","","","NULLABLE","REQUIRED",""]
    fieldlist = [fielddef_nameidws, fielddef_relOK]

    for wsFC in wsfclist:

        count = count + 1

        for mylist in fieldlist:
            if len(self.gp.ListFields(wsFC,mylist[0]))==0:
                self.gp.AddField_management(wsFC,
ist[0],mylist[1],mylist[2],mylist[3],mylist[4],mylist[5],mylist[6],mylist[7],mylist[8])

                myfcname = os.path.basename(wsFC)
                nameidbase = string.replace(myfcname.lower(), "_ws", "")
                nameidsw = nameidbase + "_sw"
                self.gp.CalculateField_management(wsFC, "nameidws", "" + nameidsw + "", "PYTHON", "")

                fcname = self.gebiet + self.wald_string + "_seg_" + "large"
                wsFCout = self.tar_gdb + "/" + fcname
                if count == 1:
                    scriptmsg = "Transfer Einzugsgebiete large in FC: " + os.path.basename(wsFCout)
                    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
                    self.cleanup(wsFCout)
                    self.gp.CreateFeatureclass(self.tar_gdb, fcname, "POLYGON", wsFC)
                    appendexpr = wsFC
                else:
                    appendexpr = appendexpr + "; " + wsFC

        self.gp.Append_management(appendexpr, wsFCout, "NO_TEST")

        tmpFCrel = self.tar_gdb + "/x_intersect"
        self.cleanup(tmpFCrel)
        intersectstring = self.srcFC_rel + ";" + wsFCout
        self.gp.Intersect_analysis(intersectstring, tmpFCrel)

    # In einem update cursor werden die rel Flaechen mit korrespondierenden Namen
    # zu wsFC markiert
    scriptmsg = "markiere zugehoerige rel in " + os.path.basename(tmpFCrel)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

```

```

rows = self.gp.UpdateCursor(tmpFCrel)
row = rows.Next()
while row:
    # Eintraege nameid auslesen
    # in nameid von rel steht nameid zu rel Layer
    nid1 = row.GetValue("nameid")
    name1 = string.replace(nid1.lower(), "_" + self.rel_string.lower(), "")
    nid2 = row.GetValue("nameidws")
    name2 = string.replace(nid2.lower(), "_sw", "")
    #print name1 + " " + name2
    if name1 == name2:
        row.relOK = "1"
    rows.UpdateRow(row)
    row = rows.Next()
del row, rows

# Reduktion auf zulaessige rel Flaechen
scriptmsg = "Reduktion auf zulaessige rel Flaechen "
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
name = "x" + self.gebiet + self.wald_string + "_sw_" + "large"
#print name
swFC = self.tar_gdb + "/" + name
self.cleanup(swFC)
selexpr = "relOK" + " = " + "1"
self.gp.FeatureClassToFeatureClass_conversion(tmpFCrel, self.tar_gdb, name, selexpr)

name = self.gebiet + self.wald_string + "_sw_" + "large"
outFCrel = self.tar_gdb + "/" + name
self.cleanup(outFCrel)
self.gp.Dissolve_management(swFC, outFCrel, "nameid", "", "SINGLE_PART")
scriptmsg = "schreibe Ergebnis " + os.path.basename(outFCrel)
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

self.cleanup(swFC)
self.cleanup(tmpFCrel)
self.cleanup(wsFCout)

return

# -----
def PROC_NAMEIDS(self):

    nameid_base = string.replace(self.nameid.lower(), "_" + self.ph_string.lower(), "")
    capname = nameid_base + self.cap_string

    # ph Features zu nameid separieren
    anzpolys = 0
    fc_ph = self.tar_gdb + "/" + self.ph_string + "_" + self.nameid
    self.cleanup(fc_ph)
    scriptmsg = "berechne " + os.path.basename(fc_ph)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
    selExpr1 = "NAMEID" + " = " + "" + "" + self.nameid + ""
    self.gp.FeatureClassToFeatureClass_conversion(self.srcFC_ph, self.tar_gdb, os.path.basename(fc_ph),
selExpr1)
    obj_count = self.gp.GetCount_management(fc_ph)
    anzpolys = int(obj_count.GetOutput(0))
    if anzpolys == 0:
        scriptmsg = "Abbruch! PH nameid: " + self.nameid + ", kein ph Poly selektiert "
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # cap Features zu nameid separieren
    anzpts = 0
    fc_cap = self.tar_gdb + "/" + self.cap_string + "_" + self.nameid
    self.cleanup(fc_cap)
    scriptmsg = "berechne " + os.path.basename(fc_cap)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
    selExpr2 = "NAMEID" + " = " + "" + "" + capname + ""
    self.gp.FeatureClassToFeatureClass_conversion(self.srcFC_cap, self.tar_gdb, os.path.basename(fc_cap),
selExpr2)
    obj_count = self.gp.GetCount_management(fc_cap)
    anzpts = int(obj_count.GetOutput(0))

```

```

if anzpts == 0:
    scriptmsg = "Abbruch! PH nameid: " + self.nameid + ", keine cap Punkte mit NAMEID = " + capname + "
gefunden."
    pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
    sys.exit()

self.gp.extent = self.setextent(fc_ph)
self.gp.snapraster = self.dtm

# create TIN

scriptmsg = "Tin " + nameid_base
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
xoutTin = self.rds_folder + "/" + "xtin"
self.cleanup(xoutTin)
self.gp.createtin_3d(xoutTin)
# edit TIN
editstring = fc_cap + " " + "ANGLE" + " " + "<none> masspoints false"
self.gp.EditTin_3d(xoutTin, editstring)

scriptmsg = "Rasterverarbeitung " + nameid_base
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

zoneGD = self.rds_folder + "/" + "zgd"
self.cleanup(zoneGD)
#self.gp.FeatureToRaster_conversion(fc_ph,"nameid",zoneGD,"10")
self.gp.FeatureToRaster_conversion(fc_ph,"objectid",zoneGD,"10")

# minimal benoetigten Extent festlegen

self.gp.extent = self.setextent(zoneGD)
self.gp.snapraster = self.dtm

xTinRaster = self.rds_folder + "/" + nameid_base
self.cleanup(xTinRaster)
self.gp.TinRaster_3D(xoutTin, xTinRaster, "FLOAT", "LINEAR", "CELLSIZE 10", "1")

"""

rgrp = "rgrp"
self.cleanup(rgrp)
#self.gp.RegionGroup_sa(zoneGD, rgrp, "EIGHT", "WITHIN", "NOLINK")
self.gp.RegionGroup_sa(zoneGD, rgrp, "EIGHT")

"""

AspectRange = self.rds_folder + "/" + "AspectRange"
self.cleanup(AspectRange)
#self.gp.ZonalStatistics_sa(fc_ph, "nameid", Aspect, AspectRange, "RANGE", "DATA")
self.gp.ZonalStatistics_sa(zoneGD, "value", self.Aspect, AspectRange, "RANGE", "DATA")

x = self.rds_folder + "/" + "zmAcosr"
self.cleanup(x)
#self.gp.ZonalStatistics_sa(fc_ph, "nameid", aspCosRad, x, "MEAN", "DATA")
#self.gp.ZonalStatistics_sa(rgrp, "value", self.aspCosRad, x, "MEAN", "DATA")
self.gp.ZonalStatistics_sa(zoneGD, "value", self.aspCosRad, x, "MEAN", "DATA")

y = self.rds_folder + "/" + "zmAsinr"
self.cleanup(y)
#self.gp.ZonalStatistics_sa(fc_ph, "nameid", aspSinRad, y, "MEAN", "DATA")
#self.gp.ZonalStatistics_sa(rgrp, "value", aspSinRad, y, "MEAN", "DATA")
self.gp.ZonalStatistics_sa(zoneGD, "value", self.aspSinRad, y, "MEAN", "DATA")

rx = self.rds_folder + "/" + "rx"
self.cleanup(rx)
self.gp.Power_sa(x, 2, rx)

ry = self.rds_folder + "/" + "ry"
self.cleanup(ry)
self.gp.Power_sa(y, 2, ry)

rs = self.rds_folder + "/" + "rs"

```

```

self.cleanup(rs)
self.gp.Plus_sa(rx,ry, rs)

r = self.rds_folder + "/" + "r"
self.cleanup(r)
self.gp.SquareRoot_sa(rs, r)

xr = self.rds_folder + "/" + "xr"
self.cleanup(xr)
self.gp.Divide_sa(x, r, xr)

yr = self.rds_folder + "/" + "yr"
self.cleanup(yr)
self.gp.Divide_sa(y, r, yr)

acosRad = self.rds_folder + "/" + "acosRad"
self.cleanup(acosRad)
self.gp.ACos_sa(xr, acosRad)

acosDeg = self.rds_folder + "/" + "acosDeg"
self.cleanup(acosDeg)
self.gp.Times_sa(acosRad,57.2957795,acosDeg)

avgDeg = self.rds_folder + "/" + "avgDeg"
self.cleanup(avgDeg)
condition = "CON (" + xr + " > 0 AND " + yr + " > 0, " + acosDeg + ", " + yr + " > 0 AND " + xr + "< 0, " +
acosDeg + ", " + xr + "< 0 AND " + yr + "< 0, 360 - " + acosDeg + ", 360 - " + acosDeg + ")"
self.gp.SingleOutputMapAlgebra_sa(condition,avgDeg)

MeanAspM90 = self.rds_folder + "/" + "MeanAspM90"
self.cleanup(MeanAspM90)
self.gp.Minus_sa(avgDeg,"90",MeanAspM90)

MeanAspP90 = self.rds_folder + "/" + "MeanAspP90"
self.cleanup(MeanAspP90)
self.gp.Plus_sa(avgDeg,"90",MeanAspP90)

OpLT180 = self.rds_folder + "/" + "OpLT180"
self.cleanup(OpLT180)
condition = "CON(" + self.Aspect + " < " + MeanAspM90 + " OR " + self.Aspect + " > " + MeanAspP90 + " ,
1, 0)"
self.gp.SingleOutputMapAlgebra_sa(condition,OpLT180)

AspRangeGT180 = self.rds_folder + "/" + "AspRangeGT180"
self.cleanup(AspRangeGT180)
self.gp.Con_sa(AspectRange,self.Aspect,AspRangeGT180,"","VALUE > 180")

AspShift360 = self.rds_folder + "/" + "AspShift360"
self.cleanup(AspShift360)
self.gp.Plus_sa(AspRangeGT180,"360", AspShift360)

MeanAspP270 = self.rds_folder + "/" + "MeanAspP270"
self.cleanup(MeanAspP270)
self.gp.Plus_sa(avgDeg,"270",MeanAspP270)

OpGE180 = self.rds_folder + "/" + "OpGE180"
self.cleanup(OpGE180)
condition = "CON(" + AspShift360 + " < " + MeanAspP90 + " OR " + AspShift360 + " > " + MeanAspP270 +
", 0, 1)"
self.gp.SingleOutputMapAlgebra_sa(condition, OpGE180)

OppSlope = self.rds_folder + "/" + "OppSlope"
self.cleanup(OppSlope)
self.gp.Con_sa(AspectRange,OpLT180,OppSlope,OpGE180,"VALUE < 180")

# ALLOCATE MEANDIRECTION of MAIN SLOPE to OPPOSITESLOPE
MainSlp = self.rds_folder + "/" + "MainSlp"
self.cleanup(MainSlp)
self.gp.SetNull_sa(OppSlope, zoneGD, MainSlp, "value = 1")

#Calc MeanDir in MainSlp
MMeanSlpDir = self.rds_folder + "/" + "MMeanSlpDir"
self.cleanup(MMeanSlpDir)
self.gp.ZonalStatistics_sa(MainSlp, "Value", self.mainSlpDir, MMeanSlpDir, "Mean", "DATA")

MMeanSlpDir05 = self.rds_folder + "/" + "MMeanSlpDir05"
self.cleanup(MMeanSlpDir05)

```



```

self.gp.Plus_sa(MMeanSlpDir,"0.5",MMeanSlpDir05)

iMMeanSlpDir = self.rds_folder + "/" + "iMMeanSlpDir"
self.cleanup(iMMeanSlpDir)
self.gp.int_sa(MMeanSlpDir05,iMMeanSlpDir)

MainMeanDir = self.rds_folder + "/" + "MainMeanDir"
self.cleanup(MainMeanDir)
reclassifyRanges = "0 1.5 1;1.5 3 2;3 6 4;6 12 8;12 24 16;24 48 32; 48 96 64;96 192 128"
self.gp.Reclassify_sa(iMMeanSlpDir, "Value", reclassifyRanges, MainMeanDir, "DATA")

#oppSlp NoData, allocate value = 1, Main Value = 0
MainOpp = self.rds_folder + "/" + "MainOpp"
self.cleanup(MainOpp)
self.gp.IsNull_sa(MainSlp, MainOpp)

NibDir = self.rds_folder + "/" + "NibDir"
self.cleanup(NibDir)
self.gp.Con_sa(MainOpp, MainMeanDir, NibDir, MainOpp,"value = 0")

oppSlpMainDir = self.rds_folder + "/" + "oppSlpMainDir"
self.cleanup(oppSlpMainDir)
self.gp.Nibble_sa(NibDir, MainMeanDir, oppSlpMainDir)

#if not OppSlope, then mainSlpDir, else oppSlpMainDir
slopeDir = self.rds_folder + "/" + "slopeDir"
self.cleanup(slopeDir)
self.gp.Con_sa(OppSlope, self.mainSlpDir, slopeDir, oppSlpMainDir,"value = 0")

#Determine flowdirection
pressDir = self.rds_folder + "/" + "pressDir"
self.cleanup(pressDir)
self.gp.FlowDirection_sa(xTinRaster,pressDir)

directionGd = self.rds_folder + "/" + "directionGd"
self.cleanup(directionGd)
condition = "CON(" + zoneGD + " > 0, CON( " + self.Aspect + " < 0 OR " + self.Slope + " < 5," + pressDir +
", " + slopeDir + ")")
self.gp.SingleOutputMapAlgebra_sa(condition, directionGd)

#-----Mask-----
self.gp.Mask = zoneGD

scriptmsg = "expand, ws " + nameid_base
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

#-----EXPAND srcFC_ph for ws-----
SPgd0 = self.rds_folder + "/" + "SPgd0"
self.cleanup(SPgd0)
self.gp.Con_sa(self.SPgd,"88",SPgd0,"","VALUE > -1")

try:
    SPgdExpand = self.rds_folder + "/" + "SPgdExpand"
    self.cleanup(SPgdExpand)
    self.gp.Expand_sa(SPgd0,SPgdExpand,"1","88")

    wList = [self.rds_folder + "/" + 'ws']
    eList = [self.rds_folder + "/" + 'exp']

    cList = range(5)
    wcList = [(a+str(b)) for a in wList for b in cList]
    ecList = [(a+str(b)) for a in eList for b in cList]

    ecList[0] = (SPgdExpand)
    ecList.append(self.rds_folder + "/" + 'exp5')

    for i in cList:
        self.gp.Watershed_sa(directionGd, ecList[i], wcList[i])
        self.gp.Expand_sa(wcList[i], ecList[i+1], "1", "88")

    actLayer = self.tar_gdb + "/" + nameid_base + "_ws"
    self.cleanup(actLayer)

    self.gp.RasterToPolygon_conversion(wcList[4], actLayer, "simplify")
    scriptmsg = "Einzugsgebiete zu " + self.nameid + " berechnet: " + os.path.basename(actLayer)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

```

```

except:
    scriptmsg = "Es liegen keine Zellen des Schadenpotentials-Rasters innerhalb "+ Expression + "!"
    pp_util_GIS.gpmsg(self.gp,scriptmsg,1)

self.cleanup(xoutTin)
self.cleanup(xTinRaster)
self.cleanup(fc_ph)
self.cleanup(fc_cap)

return
# -----
def GET_NAMEIDS(self):

    # Liste der ph Features fuer die Prozesssteuerung
    nameid_list = []
    rows = self.gp.SearchCursor(self.srcFC_ph)
    row = rows.Next()
    while row:
        # Eintraege Prozessart auslesen
        nameid = row.GetValue("nameid")
        if nameid not in nameid_list:
            nameid_list.append(nameid)
        row = rows.Next()
    del row, rows

    S_file = open(self.nameidfile, "w")
    for nameid in nameid_list:
        S_file.write(nameid + "\n")
    S_file.close()

    return

# -----
def RDS_BASIS_GEBIET(self):
    scriptmsg = "berechne Rastergrundlagen zum Extent von " + os.path.basename(self.srcFC_ph)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

    # Extent festlegen
    self.gp.extent = self.setextent(self.srcFC_ph)
    self.gp.snapraster = self.dtm

    self.cleanup(self.Aspect)
    self.gp.Aspect_sa(self.dtm, self.Aspect)

    self.cleanup(self.aspRad)
    self.gp.Times_sa(self.Aspect,0.017453293,self.aspRad)

    self.cleanup(self.aspCosRad)
    self.gp.Cos_sa(self.aspRad,self.aspCosRad)

    self.cleanup(self.aspSinRad)
    self.gp.Sin_sa(self.aspRad,self.aspSinRad)

    self.cleanup(self.Slope)
    self.gp.Slope_sa(self.dtm,self.Slope)

    self.cleanup(self.mainSlpDir)
    self.gp.FlowDirection_sa(self.dtm,self.mainSlpDir)

    self.cleanup(self.SPgd)
    self.gp.FeatureToRaster_conversion(self.srcFC_spot,"Shape_Area",self.SPgd,self.Aspect)

    self.cleanup(self.aspRad)

    return

# -----
def PROC_SPOTREL_PH(self):
    # -----> Auswertung schadenpotenzialsrelevante ph (Prozessraeume) zu LARGE

    self.pp_proc_spotrel_ph()

```

```

if not self.gp.exists(self.srcFC_ph):
    scriptmsg = "Abbruch! Grundlage fehlt: " + self.srcFC_ph
    pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
    sys.exit()

if not self.gp.exists(self.srcFC_rel):
    scriptmsg = "Abbruch! Grundlage fehlt: " + self.srcFC_rel
    pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
    sys.exit()

return

#-----
def BASISDATEN(self):
    # -----> Bereitstellung der zu verarbeitenden Datenstrukturen zum Gebiet
    # -----> Die Datenstrukturen in src_Workspace sind Coverages

    self.gp.workspace = self.src_Workspace
    self.gp.scratchworkspace = self.src_Workspace

    # cap-Datenquelle in tar_gdb kopieren (Punkt-Angaben zu Lawinen-Drucken)
    scriptmsg = "berechne " + os.path.basename(self.srcFC_cap)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
    srcFC_cap = self.pp_proc_cap()
    self.cleanup(self.srcFC_cap)
    self.gp.CopyFeatures(srcFC_cap,self.srcFC_cap)

    # Schadenpotenzial in tar_gdb importieren,
    scriptmsg = "berechne " + os.path.basename(self.srcFC_spot)
    pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
    expression = ""
    self.cleanup(self.srcFC_spot)
    self.gp.FeatureClassToFeatureClass_conversion(self.DamLayer, self.tar_gdb,
os.path.basename(self.srcFC_spot), expression)

#-----
def CREATE_WORKSPACES(self):
    # -----> self.tar_gdb erstellen
    try:
        self.cleanup(self.tar_gdb)
        self.gp.CreateFileGDB_management(self.tar_Folder, self.tar_gdbname)

        if self.gp.exists(self.tar_gdb):
            scriptmsg = "Ablage Ergebnisse in: " + self.tar_gdb
            pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
    except:
        scriptmsg = "Abbruch: " + self.tar_gdbname + " kann in " + self.tar_Folder + " nicht erstellt werden!"
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # -----> self.rds_folder fuer TIN und Raster-Zwischenergebnisse erstellen
    if not os.path.isdir(self.rds_folder):
        try:
            os.mkdir(self.rds_folder)
        except:
            scriptmsg = "Abbruch: " + os.path.basename(self.rds_folder) + " kann in " + self.tar_Folder + " nicht
erstellt werden!"
            pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
            sys.exit()
    return

#-----
def cleanup(self,infc):
    if self.gp.exists(infc):
        self.gp.delete(infc)
    return

# -----
def pp_proc_cap(self):

    mycapFClist = []

```

```

self.gp.workspace = self.src_Workspace
searchstring = self.gebiet.lower() + "n*_cap"
myWSlist = self.gp.ListWorkspaces(searchstring)

if len(myWSlist)>0:

    for mycapFDS in myWSlist:
        self.gp.workspace = mycapFDS
        mycapFCs = self.gp.ListFeatureClasses(".*", "Point")
        for srcFC_cap in mycapFCs:
            #print os.path.basename(srcFC_cap)
            mycapFClist.append(srcFC_cap)

    if len(mycapFClist)<>1:
        scriptmsg = "Abbruch! Zu gebiet " + self.gebiet + " sind mehrere/keine cap Datenquellen definiert!"
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    return srcFC_cap

# -----
def bildeFC_Paare_ph_rel(self,outFCs):

    # FC Liste filtern auf Feature Klassen mit Gebietsname und ph_string oder rel_string
    theFCs = []
    for myFC in outFCs:
        myfdsname = os.path.dirname(myFC)
        mydirname = os.path.dirname(myfdsname)

        afcname = string.replace(myFC.lower(), mydirname.lower(), "")
        for ffix in [self.ph_string, self.rel_string]:
            if afcname.find(self.gebiet) >= 0:
                if afcname.find(ffix) >= 0:
                    theFCs.append(myFC)

    # FC Liste filtern auf Feature Klassen mit Feldern nameid
    keepFCs = []
    fieldlist = ["nameid", "id"]
    for myFC in theFCs:
        anzfound = 0
        for searchfieldname in fieldlist:

            if len(self.gp.ListFields(myFC,searchfieldname))>0:
                anzfound = anzfound + 1

        if anzfound == len(fieldlist):
            keepFCs.append(myFC)

    # ph und rel FC Listen aufbauen
    ph_list = []
    rel_list = []
    for myFC in keepFCs:
        myfdsname = os.path.dirname(myFC)
        mydirname = os.path.dirname(myfdsname)

        afcname = string.replace(myFC.lower(), mydirname.lower(), "")
        if afcname.find(self.ph_string) >= 0:
            ph_list.append(myFC)
        elif afcname.find(self.rel_string) >= 0:
            rel_list.append(myFC)

    # zusammengehoerige Paare bilden
    srcPairs = []
    for phFC in ph_list:
        afcnameph = string.replace(phFC.lower(), self.ph_string.lower(), "")
        #print afcnameph
        for relFC in rel_list:
            afcnamerel = string.replace(relFC.lower(), self.rel_string.lower(), "")
            #print afcnamerel
            if (afcnameph == afcnamerel):
                srcPairs.append([phFC,relFC])

    anzpairs = len(srcPairs)

```

```

scriptmsg = str(len(srcPairs)) + " gueltige ph / rel FC Paare zu Gebiet " + self.gebiet
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

return srcPairs

# -----
def procFDSlist(self,myFDSlist):

    outFCs = []
    myFClist = []
    for myFDS in myFDSlist:

        self.gp.workspace = myFDS
        myFCs = self.gp.ListFeatureClasses("*. *", "Region")
        for myFC in myFCs:
            FullFC = myFDS + "/" + myFC
            if self.gp.exists(FullFC):
                myFClist.append(FullFC)

    if len(myFClist)>0:

        for fFC in myFClist:
            dsc = self.gp.describe(fFC)
            # wenn's eine FAT hat, ists ein Coverage
            if dsc.HasFAT:
                if dsc.FeatureClassType == "Region":
                    outFCs.append(fFC)
            # kein Coverage
            else:
                if dsc.FeatureClassType == "Region":
                    outFCs.append(fFC)

    return outFCs

# -----
def pp_proc_spotrel_ph(self):

    self.gp.workspace = self.src_Workspace
    searchstring = self.gebiet.lower() + ".n*"
    myWSlist = self.gp.ListWorkspaces(searchstring)
    myFDSlist = []
    for entry in myWSlist:
        self.gp.workspace = entry
        myFClist = self.gp.ListFeatureClasses("*. *", "REGION")
        if len(myFClist)>0:
            myFDSlist.append(entry)

    if len(myFDSlist)==0:
        scriptmsg = "Abbruch: keine gueltigen Regions in " + self.src_Workspace
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    # FC Liste erstellen

    srcPairs = []

    outFCs = self.procFDSlist(myFDSlist)

    # self.gebietsweise ph und rel FC Paare pruefen.
    if len(outFCs) > 0:
        srcPairs = self.bildeFC_Paare_ph_rel(outFCs)

    self.gp.ScratchWorkspace = self.tar_gdb
    self.gp.workspace = self.tar_gdb

    # wenn self.gebietsweise ph und rel Paare vorhanden...
    if len(srcPairs) > 0:

```

```

# ph's zusammenfuehren in Ziel Feature Klasse in tar_Folder
# beinhaltet alle Features ph, bewaldet und waldfrei
expression = ""
phFC = ""
for pair in srcPairs:
    phFC = pair[0]
    if len(expression) == 0:
        expression = phFC
    else:
        expression = expression + ";" + phFC
tar_PHall = "x_" + self.ph_string + "_all"
self.cleanup(tar_PHall)
self.gp.Merge_management(expression, tar_PHall, "NO_TEST")

# von den phs wird nur range_code 2 und 4 verwendet.
# code 0 liegt ausserhalb des Lawinenperimeters
tar_PH = "x_phc"
self.cleanup(tar_PH)
selexpr = "range_code" + " <> " + "0"
self.gp.Select_analysis(tar_PHall, tar_PH, selexpr)
# dissolve der Lawinenzuege auf nameid
tar_phFC = "x_" + self.ph_string
self.cleanup(tar_phFC)
self.gp.Dissolve_management(tar_PH, tar_phFC, "nameid", "", "SINGLE_PART")
self.cleanup(tar_PHall)
self.cleanup(tar_PH)

# rel's zusammenfuehren in Ziel Feature Klasse in tar_Folder
# beinhaltet alle Features rel, bewaldet und waldfrei
expression = ""
for pair in srcPairs:
    relFC = pair[1]
    if len(expression) == 0:
        expression = relFC
    else:
        expression = expression + ";" + relFC
tar_relFC = "x_" + self.rel_string
self.cleanup(tar_relFC)
self.gp.Merge_management(expression, tar_relFC, "NO_TEST")

#Make Feature Layer Damage
MemDamage = "MemDamage"
self.cleanup(MemDamage)
self.gp.MakeFeatureLayer(self.srcFC_spot, MemDamage)

for size in ["s", "m", "l"]:
    scriptmsg = "verarbeite Features Kategorie: " + size + ", gebiet: " + self.gebiet
    pp_util_GIS.gpmsg(self.gp, scriptmsg, 0)

    sizetext = ""

    if size == "s":
        sizetext = "small"
    if size == "m":
        sizetext = "medium"
    if size == "l":
        sizetext = "large"

    ph_namelist = []
    rel_namelist = []

# ----- VERARBEITUNG Schadenpotenzialsrelevanz PH
# ph_namelist fuer Groessenkategorie erstellen,
# wegfiltern Eintraege n (Nichtwald)

# Featurelayer ph anlegen
MemPH = "MemPH"
self.cleanup(MemPH)
self.gp.MakeFeatureLayer(tar_phFC, MemPH)

```



```

# Featurelayer rel anlegen
MemREL = "MemREL"
self.cleanup(MemREL)
self.gp.MakeFeatureLayer(tar_relFC, MemREL)

# Selektion der Features n, self.gebiet und Groessenklasse
anzPHsize = 0
anzPHspot = 0
selstring = self.gebiet + self.wald_string + "_" + size + "%"
Exp_ph = "NAMEID LIKE '" + selstring + "'"
scriptmsg = "Selektiere PH " + selstring
pp_util_GIS.gpmsg(self.gp, scriptmsg, 0)
self.gp.SelectLayerByAttribute(MemPH, "NEW_SELECTION", Exp_ph)
obj_count = self.gp.GetCount_management(MemPH)
anzPHsize = int(obj_count.GetOutput(0))

# Reduktion auf schadenpotenzialsrelevante ph Flaechen
if anzPHsize > 0:
    scriptmsg = "Selektiere PH spot relevante"
    pp_util_GIS.gpmsg(self.gp, scriptmsg, 0)

self.gp.SelectLayerByLocation_Management(MemPH, "INTERSECT", MemDamage, "", "SUBSET_SELECTION")
obj_count = self.gp.GetCount_management(MemPH)
anzPHspot = int(obj_count.GetOutput(0))
scriptmsg = "in PH " + self.wald_string + ", Gebiet " + self.gebiet + ", " + size + ": " +
str(anzPHsize) + " Features"
pp_util_GIS.gpmsg(self.gp, scriptmsg, 0)

# Resultat pro Groessenklasse schreiben
if anzPHspot > 0:
    nameout = self.gebiet + self.wald_string + "_" + self.ph_string + "_" + sizetext
    expression = ""
    outFCph = self.tar_gdb + "/" + nameout
    self.cleanup(outFCph)
    self.gp.FeatureClassToFeatureClass_conversion(MemPH, self.tar_gdb, nameout, expression)
    scriptmsg = "in PH " + self.wald_string + ", Gebiet " + self.gebiet + ", " + size + ": " +
str(anzPHsize) + " Features spot relevant"
    pp_util_GIS.gpmsg(self.gp, scriptmsg, 0)

# ----- VERARBEITUNG REL
# Die spot relevanten PH Features werden mit den rel Features verschnitten.
if anzPHspot > 0:
    nameout = "xi_" + self.rel_string
    tmpFCrel = self.tar_gdb + "/" + nameout
    self.cleanup(tmpFCrel)
    intersectstring = tar_relFC + ";" + outFCph
    self.gp.Intersect_analysis(intersectstring, tmpFCrel)
    self.gp.AddField_management(tmpFCrel,
"phOK", "SHORT", "", "", "", "NULLABLE", "REQUIRED", "")

# In einem update cursor werden die rel FlÃ¤chen in den korrekten ph FlÃ¤chen markiert
rows = self.gp.UpdateCursor(tmpFCrel)
row = rows.Next()

while row:
    # Eintraege Prozessart auslesen
    nid1 = row.GetValue("nameid")
    nid2 = row.GetValue("nameid_1")
    name1 = string.replace(nid1.lower(), self.rel_string.lower(), "")
    name2 = string.replace(nid2.lower(), self.ph_string.lower(), "")
    if name1 == name2:
        row.phOK = "1"

    rows.UpdateRow(row)
    row = rows.Next()
del row, rows

# Reduktion auf zulaessige rel FlÃ¤chen

```

```

        nameout = "x" + self.gebiet + self.wald_string + "_" + self.rel_string + "_" + sizetext
        outFCrelOK = self.tar_gdb + "/" + nameout
        self.cleanup(outFCrelOK)
        selexpr = "phOK" + " = " + "1"
        self.gp.Select_analysis(tmpFCrel, outFCrelOK, selexpr)

        obj_count = self.gp.GetCount_management(outFCrelOK)
        anzREL = int(obj_count.GetOutput(0))
        if anzREL > 0:
            scriptmsg = "in zugehoerigen REL " + self.wald_string + ", Gebiet " + self.gebiet + ", " + size +
            ": " + str(anzPHsize) + " Features"
            pp_util_GIS.gpmsg(self.gp,scriptmsg,0)

            self.cleanup(tmpFCrel)
            nameout = self.gebiet + self.wald_string + "_" + self.rel_string + "_" + sizetext
            outFCrel = self.tar_gdb + "/" + nameout
            self.cleanup(outFCrel)
            self.gp.Dissolve_management(outFCrelOK, outFCrel, "nameid", "", "SINGLE_PART")
            self.cleanup(outFCrelOK)

        self.cleanup(tar_phFC)
        self.cleanup(tar_relFC)

    else:
        scriptmsg = "Abbruch: Keine gueltigen Daten zu gebiet " + self.gebiet
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    else:
        scriptmsg = "Abbruch: Keine Feature Klassen in " + self.src_Workspace
        pp_util_GIS.gpmsg(self.gp,scriptmsg,2)
        sys.exit()

    return

# -----
def setextent(self,myextfc):
    dsc = self.gp.describe(myextfc)
    myextent = dsc.Extent

    xmin = myextent.xmin - pp_Settings.addext
    ymin = myextent.ymin - pp_Settings.addext
    xmax = myextent.xmax + pp_Settings.addext
    ymax = myextent.ymax + pp_Settings.addext

    myextent = str(xmin) + " " + str(ymin) + " " + str(xmax) + " " + str(ymax)

    return myextent

"""
scriptmsg = "Dauer der Verarbeitung: " + timeconsumstring(starttime, time.clock())
pp_util_GIS.gpmsg(self.gp,scriptmsg,0)
"""

```

### 9.3.4 Pp\_Settings.py: paramétrage et réglage

```
ï»¿  
pp_Settings.py
```

Der Prozess pp\_large setzt die Verfügbarkeit folgender Skripts voraus:

```
pp_large_launch.py (aufrufendes Skript)  
pp_large_proc_nameid.py  
pp_large_proc_rel.py  
pp_Settings.py  
pp_util_GIS.py
```

Autor: geo7 AG, Peter Gsteiger

History: 30.06.2011 programmiert fuer ArcGIS 9.3.1, Python 25

Version: 1.0

Description:  
Konfiguration Prozess pp\_large

Workflow und Produkte sind beschrieben in:  
Identifikation schadenpotenzialsrelevanter Lawinenanrissgebiete Silvaproduct  
geo7, 30.06.2011

```
'''
```

```
# -----  
# Python Modules  
# -----  
import sys, os
```

```
# -----  
# Settings  
# -----
```

```
# erforderliche ArcGIS-Lizenzierung:  
ARCLICREQ = "ArcInfo"
```

```
# Parameter:  
src_Workspace = r"D:\SilvaProtect\BASIC\INPUT\WIL\loadFC"  
tar_Folder = r"D:\SilvaProtect\SZENARIEN\SBB\IN_urGN_Lawine"  
gebiet = "WIL"  
DamLayer = r"D:\SilvaProtect\SZENARIEN\SBB\DA_urGN_Lawine\DA_wil.shp"  
dtm = r"D:\SilvaProtect\BASIC\DHM.mdb\DHM_10"
```

```
# Installationspfad Python:  
_path_python = r"C:\Program Files (x86)\Python25"  
_path_python = r"C:\Python25"  
# Pfad Skripte:  
_path_script = r"D:\SilvaProtect\INTERSECT\LAWINE_geo7\20110704_BAFU\Scripts"
```

```
# Zuschlag [m] Mapextent bei Rasterverarbeitungen  
addext = 500
```

```
# Variablen zum nameid handling  
# !!!!!Eintrage nicht veraendern!!!!  
ph_string = "ph"  
rel_string = "rel"  
cap_string = "cap"  
wald_string = "n"
```

### 9.3.5 Pp\_util\_GIS.py: appel objets SIG et avertissements

ï»¿  
pp\_util\_GIS.py

Der Prozess pp\_large setzt die Verfügbarkeit folgender Skripts voraus:

pp\_large\_launch.py (aufrufendes Skript)  
pp\_large\_proc\_nameid.py  
pp\_large\_proc\_rel.py  
pp\_Settings.py  
pp\_util\_GIS.py

Autor: geo7 AG, Peter Gsteiger

History: 30.06.2011 programmiert fuer ArcGIS 9.3.1, Python 25

Version: 1.0

Description:

Bezug gp-Objekt, Einstellungen zur Geoverarbeitung, Benachrichtigungen

Workflow und Produkte sind beschrieben in:

Identifikation schadenpotenzialsrelevanter Lawinenanrissgebiete Silvaproduct  
geo7, 30.06.2011

'''

```
# -----  
# Python Modules  
# -----  
import sys, os, string,time
```

```
# -----  
def gdbsettings(gp):
```

```
    gp.outputCoordinateSystem =  
"PROJCS['CH1903_LV03',GEOGCS['GCS_CH1903',DATUM['D_CH1903',SPHEROID['Bessel_1841',6377397.  
155,299.1528128]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Hotine_Obli  
que_Mercator_Azimuth_Center'],PARAMETER['False_Easting',600000.0],PARAMETER['False_Northing',20000  
0.0],PARAMETER['Scale_Factor',1.0],PARAMETER['Azimuth',90.0],PARAMETER['Longitude_Of_Center',7.439  
583333333333],PARAMETER['Latitude_Of_Center',46.9524055555556],UNIT['Meter',1.0]]"
```

```
    gp.XYResolution = "0.001 Meters"  
    gp.MResolution = "0.001"  
    gp.ZResolution = "0.001 Meters"  
    gp.XYTolerance = "0.002 Meters"  
    gp.MTolerance = "0.002"  
    gp.ZTolerance = "0.002 Meters"
```

```
    gp.randomGenerator = "0 ACM599"  
    gp.outputZFlag = "Same As Input"  
    gp.outputMFlag = "Same As Input"  
    gp.outputZValue = ""  
    gp.qualifiedFieldNames = "true"  
    gp.extent = "DEFAULT"  
    gp.geographicTransformations = ""
```

```
    return
```

```
# -----  
def GIS_Setup(ARCLICREQ):
```

```
    gp = None  
    scriptmsg = ""
```

```
    # Geoprocessor Objekt erstellen
```

```
    try:
```

```
        import arcgisscripting  
        gp = arcgisscripting.create(9.3)
```

```
    except:
```

```
        scriptmsg = "Abbruch: Das Geoprocessor Objekt konnte nicht erstellt werden!"  
        gpmsg(gp,scriptmsg,2)  
        sys.exit()
```

```

# Liste alle Lizenzen
licenses_all = []
licenses_all.append("ArcEditor")
licenses_all.append("ArcView")
licenses_all.append("ArcInfo")

# bestimmen verfügbare Lizenzen
licenses_available = []
rsp_suit = ""
for lic in licenses_all:
    rsp_suit = gp.CheckProduct(lic)
    #scriptmsg = "CheckProduct fuer " + lic + " hat Status " + str(rsp_suit)
    #gpmsg(gp,scriptmsg,0)
    if rsp_suit in ("Available", "AlreadyInitialized"):
        licenses_available.append(lic)

# bestimmen taugliche Lizenzen
licenses_suitable = []
if string.lower(ARCLICREQ) == string.lower("ArcView"):
    licenses_suitable.append("ArcView")
    licenses_suitable.append("ArcEditor")
    licenses_suitable.append("ArcInfo")
elif string.lower(ARCLICREQ) == string.lower("ArcEditor"):
    licenses_suitable.append("ArcEditor")
    licenses_suitable.append("ArcInfo")
elif string.lower(ARCLICREQ) == string.lower("ArcInfo"):
    licenses_suitable.append("ArcInfo")

# bestimmen Schnittmenge verfügbarer und tauglicher Lizenzen
licenses_needed = []
for lic_t in licenses_suitable:
    for lic_v in licenses_available:
        if lic_t == lic_v:
            licenses_needed.append(lic_t)
if len(licenses_needed) == 0:
    scriptmsg = "Es ist keine Lizenz " + ARCLICREQ + " verfügbar."
    gpmsg(gp,scriptmsg,2)
    sys.exit()

# aktive Lizenz (Desktop Manager) bestimmen
try:
    gp.SetProduct(ARCLICREQ)
except:
    scriptmsg = "Abbruch! Desktop Manager Lizenz korrigieren auf " + str(ARCLICREQ)
    gpmsg(gp,scriptmsg,2)
    sys.exit()

# Load required Extensions...
try:
    gp.CheckOutExtension("spatial")
except:
    scriptmsg = "Abbruch: Die Spatial Extension ist nicht verfügbar."
    gpmsg(gp,scriptmsg,2)
    sys.exit()

# Load required Extensions...
try:
    gp.CheckOutExtension("3D")
except:
    scriptmsg = "Abbruch: Die 3D Extension ist nicht verfügbar."
    gpmsg(gp,scriptmsg,2)
    sys.exit()

return gp

# -----
def gpmsg(gp,message,sev):

    yr = time.strftime("%Y")

```

```
mt = time.strftime("%m")
d = time.strftime("%d")
hr = time.strftime("%H")
mte = time.strftime("%M")
sec = time.strftime("%S")
timestring = yr + "_" + mt + "_" + d + " " + hr + ":" + mte + ":" + sec

message = str(timestring) + " " + message
```

```
try:
    if sev == 2:
        gp.AddError(message)
    elif sev == 1:
        gp.AddWarning(message)
    else:
        gp.AddMessage(message)
    print message
except:
    print message
```

```
return
```



## 9.4 Temps de calcul

| Modules                                               | Temps estimé |
|-------------------------------------------------------|--------------|
| Enjeux                                                | 5 h          |
| Avalanches (INTERSECT)                                | 100 h        |
| Chutes de pierres (INTERSECT)                         | 30 h         |
| Coulée de boue (INTERSECT)                            | 50 h         |
| Cours d'eau pertinents (INTERSECT)                    | 40 h         |
| Processus pertinents pour les cours d'eau (INTERSECT) | 150 h        |
| Index (SYNTHESE)                                      | 5 h          |
| <b>Estimation totale</b>                              | <b>380 h</b> |